GEORGIA INSTITUTE OF TECHNOLOGY — OFFICE OF CONTRACT ADMINISTRATION

PROJECT ADMINISTRATION DATA SHEET

[x] ORIGINAL [ ] REVISION NO. ______

Project No. E-24-603 GTRI/~~GIT~~ DATE 2 / 20 / 84

Project Director: Dr. L.G. Callahan School/~~Lab~~ ISyE

Sponsor: U.S. Army Missile Command, Redstone Arsenal

Type Agreement: Purchase Order No. DAAH01-84-P-1319

Award Period: From 1/12/84 To 6/15/84 (Performance) 6/15/84 (Reports)

Sponsor Amount:	This Change	Total to Date
Estimated:	$ 8,200	$ 8,200
Funded:	$ 8,200	$ 8,200

Cost Sharing Amount: $ ______ Cost Sharing No: ______

Title: Interactive Computer Simulation of Urban Warfare

ADMINISTRATIVE DATA OCA Contact Lynn Boyd x4820

1) Sponsor Technical Contact:

Ms. Vera Drinnon
U.S. Army Missile Command
DRSMI-ICDB/Army Missile Lab
Redstone Arsenal, AL 35898
(205) 876-8748

2) Sponsor Admin/Contractual Matters:

Juliette F. Sperr
Contracting Officer
USA, MICOM, DIR for Proc and Prod
Attn: DRSMI-ICDB
Redstone Arsenal, AL 35898

Defense Priority Rating: DO-A2 under DMS Reg 1

Military Security Classification: n/a

(or) Company/Industrial Proprietary: n/a

RESTRICTIONS

See Attached ______ Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval – Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with gov't; however, none proposed.

COMMENTS:

RECEIVED FEB 1984 Research Reports Office

COPIES TO:

Project Director	Procurement/EES Supply Services	GTRI
Research Administrative Network	~~Research Security Services~~	Library
Research Property Management	Reports Coordinator (OCA)	Project File
Accounting	Research Communications (2)	Other NEWTON

FORM OCA 4:383

GEORGIA INSTITUTE OF TECHNOLOGY

OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION/CLOSEOUT SHEET

Date 7/18/84

Project No. E-24-603

School/~~Lab~~ ISyE

Includes Subproject No.(s)

Project Director(s) Dr. L. G. Callahan GTRI / ~~GIT~~

Sponsor U. S. Army MICOM, Redstone Arsenal, AL

Title Interactive Computer Simulation of Urban Warfare

Effective Completion Date: 6/15/84 (Performance) 6/15/84 (Reports)

Grant/Contract Closeout Actions Remaining:

- [ ] None
- [x] Final Invoice or Final Fiscal Report
- [ ] Closing Documents
- [ ] Final Report of Inventions
- [ ] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other

Continues Project No. Continued by Project No.

COPIES TO:

Project Director
Research Administrative Network
Research Property Management
Accounting
Procurement/EES Supply Services
Research Security Services
Reports Coordinator (OCA)
Legal Services

Library
GTRI
Research Communications (2)
Project File
Other I. Newton

Form OCA 60:1028

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Interactive Computer Simulation of Urban Warfare	5. TYPE OF REPORT & PERIOD COVERED Final Summary Report
	6. PERFORMING ORG. REPORT NUMBER 1/12/84-6/15/84
7. AUTHOR(s) Leslie G. Callahan Jr. and Robert A. Vrtis	8. CONTRACT OR GRANT NUMBER(s) P.O. DAAH01-84-P-1319
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Ms. Vera Drinnon, U.S. Army Missile Command, DRSMI-ICDB, Redstone Arsenal, AL 35898	12. REPORT DATE June 1984
	13. NUMBER OF PAGES 120
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Urban Warfare, Simulation, BlockBuster, MOUT, Training Games, Micro-Computers

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research proposes a methodology for the transforming of the manual war game, BLOCKBUSTER, into a micro-computer drive war game for the purpose of training junior leaders in MOUT. By identifying general attributes desired to be portrayed in such a war game, the graphic, computational and storage capabilities of a micro-computer are used to enhance the amount of realism capable of being portrayed and to reduce playing time.

CONTINUED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (continued)

This proposed methodology incorporates the suppression and building clearance algorithms of ACABUG and the digitized terrain data base and line of sight and movement algorithms of ARTBASS rather than attempting to develop new such algorithms. Additionally, the MICROFIX micro-computer system was investigated as a potential hardware base for the war game. The data base management system and the ability to overlay graphics on standard military map sheets available in MICROFIX are directly applicable to this proposed methodology.

Original research produced a means of portraying urban terrain as an overlay and providing information from this urban overlay for line of sight checks. Additionally, a mathematical model and algorithm which allows players to apply the common tactic of assigning sectors of responsibility to individual weapon systems has been developed.

The evaluation of a model is necessarily a subjective, management function. A technique for evaluating the use of this proposed war game as a training aid when compared to three other war games is explained. Additionally, the difficult question of the validity of a combat model as an accurate representation combat is explored.

Recommendations for further research into areas not addressed by this methodology are provided.

Final Summary Report for the
United States Army Missile Command
Redstone Arsenal
Huntsville, Alabama 35898

With Funding Provided by the
United States Army Combined Arms Research Activity
Fort Leavenworth, Kansas 66027

"Interactive Computer Simulation of
Urban Warfare"

Under Purchase Order No. DAAH01-84-P-1319

Conducted by

The School of Industrial and Systems Engineering
Georgia Institute of Technology

Leslie G. Callahan, Jr., Project Director
Robert A. Vrtis, Investigator

June 1984

(unclassified)

ACKNOWLEDGEMENTS

This report is principally based on Captain Robert A. Vrtis' thesis, submitted in fulfillment of the requirements for the Master of Science in Operations Research at the School of Industrial and Systems Engineering, Georgia Institute of Technology in March 1984. The research was funded by the United States Army Combined Arms Research Activity under a contract with the United States Army Missile Command. The following thesis members approved the final report:

Dr. Leslie G. Callahan Jr., Chairman

Dr. Ross A. Gagliano

Dr. Donovan B. Young

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

This research proposes a methodology for the transforming of the manual war game, BLOCKBUSTER, into a micro-computer driven war game for the purpose of training junior leaders in MOUT. By identifying general attributes desired to be portrayed in such a war game, the graphic, computational and storage capabilities of a micro-computer are used to enhance the amount of realism capable of being portrayed and to reduce playing time.

This methodology incorporates the suppression and building clearance algorithms of ACABUG and the digitized terrain data base and line of sight and movement algorithms of ARTBASS rather than attempting to develop new such algorithms. Additionally, the MICROFIX micro-computer system was investigated as a potential hardware base for the war game. The data base management system and the ability to overlay graphics on standard military map sheets available in MICROFIX are directly applicable to this proposed methodology.

Original research produced a means of portraying urban terrain as an overlay and providing information from this urban overlay for line of sight checks. Additionally, a mathematical model and algorithm which allows players to apply the common tactic of assigning sectors of responsibility to individual weapon systems has been developed.

The evaluation of a model is necessarily a subjective, management function. A technique for evaluating the use of this proposed war game as a training aid when compared to three other war games is explained. Additionally, the difficult question of the validity of a combat model as an accurate representation combat is explored.

Recommendations for further research into areas not addressed by this methodology are provided.

CHAPTER I

INTRODUCTION

Background

Combat models have long been used by the miliary as a basis for education, training, and design. At the present time, however, the defense analytical community has not adopted a common set of definitions for the classification and use of such combat models. For the purpose of this research we will accept the following definitions for modeling, simulation, and gaming proposed by Dr. L. G. Callahan [22] in 1982:

Modeling	- A specific way of expressing a theory which reveals the internal structure.
Simulation	- Experimentation with models - use of computers as tools for modeling
Gaming	- Simulations involving human operators in the conduct of an experiment.

The advent of the computer and the resultant degree to which the complexities of modern combat can be portrayed has resulted in a proliferation of computer based models, simulations, and games. Figure 1 provides graphic representation of the spectrum of types of combat models and the trade-offs which exist between the amount of operational realism portrayed in the model and the convenience, accessability, and degree of abstraction of the model.

Let us then further limit our discussion of combat models to war games. Lawrence J. Low [18] has developed a Gaming Classification

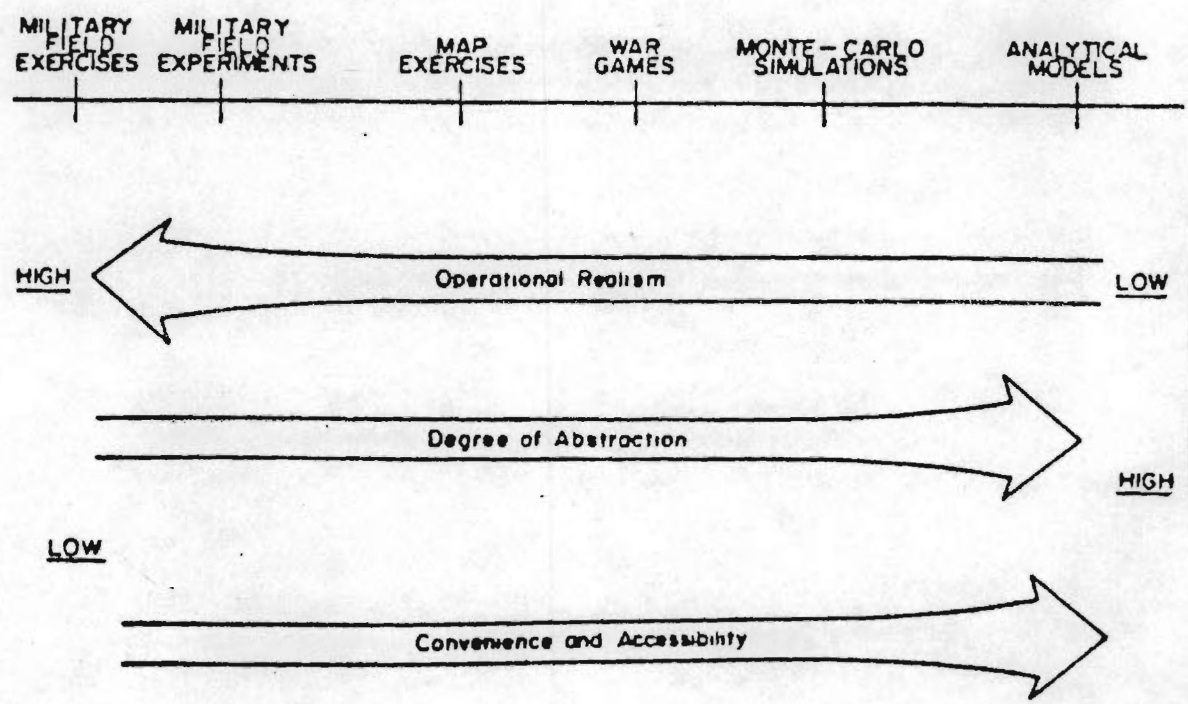


Figure 1. The Spectrum of Types of Combat Models

matrix which further assists in defining the areas which will be discussed in this thesis. This matrix, shown in Figure 2, illustrates the enormous amount of analysis for which war games are currently being used.

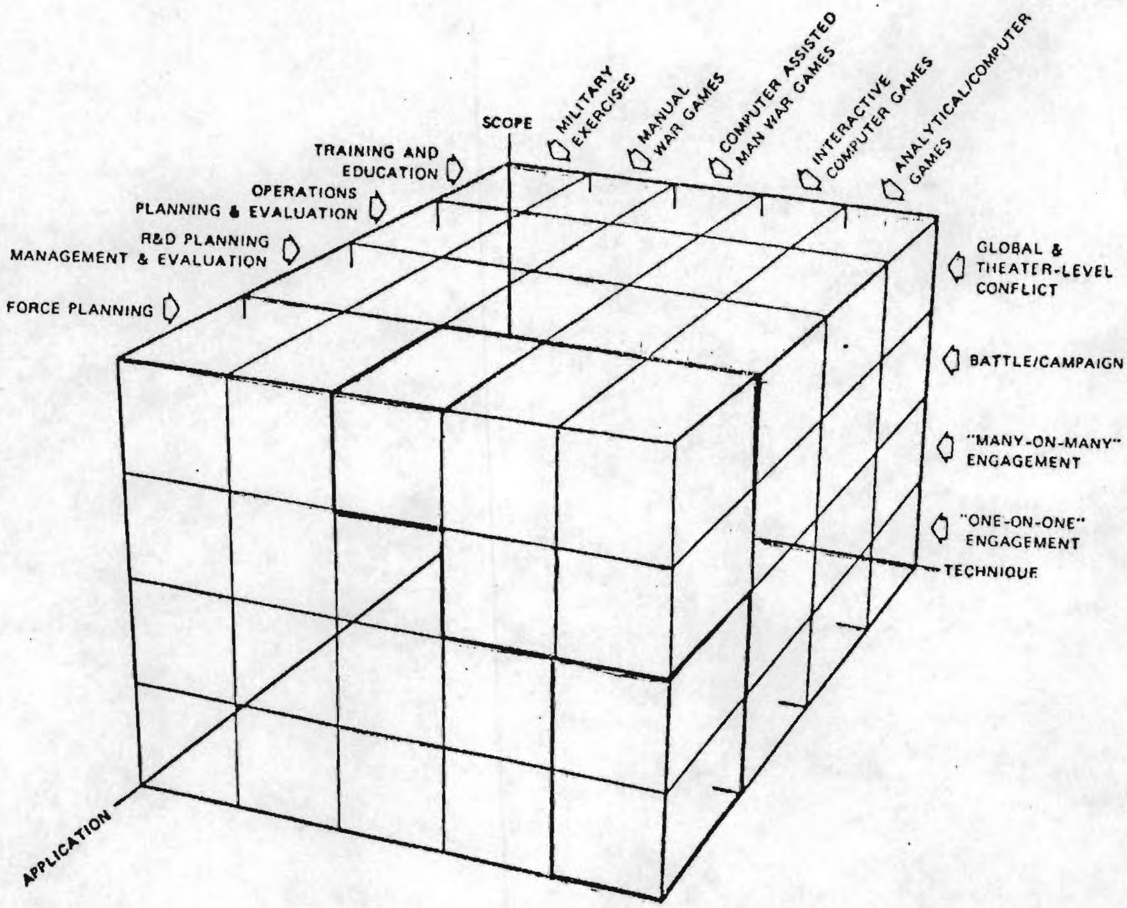


Figure 2. Gaming Classification Matrix

In the post World War II era, large scale, computer-assisted war games and interactive computer games were developed for use in force development, and research and development planning at upper echelons of the Army. This was primarily a result of the cost of hardware required to support a computer-assisted war game. Recent advances in computer technology and the reduced cost of computer systems are beginning to affect this situation. Micro and mini-computer driven war games are currently being designed to train commanders and staffs at the battalion and lower levels.

The general goal of this research is to investigate the potential use of micro and mini-computer driven war games for training junior leaders, vis a vis the standard family of manual war games. The taxonomical inventory scheme adopted the U.S. Army's TRADOC Systems Analysis Activity (TRASANA), shown in Table 1, provides a convenient frame work for defining the scope of the research conducted in this thesis. An examination of the manual models listed in Table 1 and the remaining family of models reveals that a computer driven training war game that replaces the BLOCKBUSTER manual war game as the standard war game for training junior leaders in urban combat would be a valuable addition to the family. By using data and techniques available in other members of this family of models, particularly the American Canadian Australian British Urban Game (ACABUG) and the Army Training Battle Simulation System (ARTBASS), as well as some new techniques, we will propose a methodology for accomplishing such a transformation.

Problem

The U.S. Army's FM 90-10, "Military Operations on Urbanized Terrain (MOUT)" [20], provides the basis for a growing emphasis on MOUT training by the military.

Table 1. TRASANA Model Inventory Outline

A. FORCE-ON-FORCE - HIGH RESOLUTION
1. ACABUG
2. ASARS
3. BATTLE
4. CARMONETTE
5. CASTFOREM
6. JANUS
7. STAR

B. FORCE-ON-FORCE - LOW RESOLUTION
1. CORDIVEM
2. DAME
3. FOURCE
4. JIFFY
5. QUICK SCREEN
6. VECTOR-2

C. FUNCTIONAL
1. ADAGE (ADA)
2. COMO (ADA)
3. COMO III (ADA)
4. FAST (FA)
5. TAFSM (FA)
6. CEMTACS (Commo)
7. SIMCE (Commo)
8. EMSIM (Intel)
9. ASAS PAM (Intel)
10. Intelligence/EW

D. LOGISTICS
1. ARM
2. BRESIM
3. COVERS
4. ERAMS
5. MACATAK
6. RETCOM
7. TARMS
8. APS
9. CAM-X
10. LADEN
11. LOGAM
12. TRANATAK

E. TRAINING - MANUAL
1. DUNN/KEMPF *
2. FIRST BATTLE
3. PEGASUS

F. TRAINING - COMPUTER ASSISTED
1. CATTS
2. MACE

G. TRAINING - COMPUTER DRIVEN
1. ARTBASS
2. NOT USED
3. CAMMS/CAMMS E

* BLOCKBUSTER is a refinement of DUNN/KEMPF which addresses urban terrain

> "Tactical doctrine stresses that urban combat operations are conducted only when required and that built-up areas are isolated and bypassed rather than risking a costly, time consuming operation in this difficult environment. Adherence to these precepts, though valid, is becoming increasingly difficult as urban sprawl changes the face of the battlefield. ... (Commanders) must understand the advantages and disadvantages urbanization offers and its effects on tactical operations."

The Army maintains that there is no substitute for realistic, challenging training in an actual urban environment. Given the nature of urban combat, frequently a series of small unit battles, the success of an operation depends greatly on the skill and training of the individual soldier and the initiative of the small unit leader. But urban training areas are barely available to develop the skills of the individual soldier and define his function as a part of a team.

The need to conduct field training exercises which provide company commanders and platoon leaders an opportunity to test and develop their skills in controlling and directing MOUT clashes with the necessary training of individuals and small units prior to the conduct of such exercises. Additionally, the expense of such exercises limits opportunities to train company commanders and platoon leaders in these skills to one or two times per year.

The BLOCKBUSTER training game attempts to fill this gap. But, it is a manual war game which requires time and personnel, limiting its use outside of officer basic and advanced courses. Computer-assisted and computer-driven training war games currently available do not represent urban terrain at the resolution which is required to train company commanders and platoon leaders.

Advances in micro-computer technology and the fielding of a micro-computer system to battalion sized units represents a potential expansion in the area of computer-driven training war games. An adaptation of manual and computer driven war gaming techniques to a micro-computer system would expand the applications of such a system. Additionally, it may be possible to overcome the time and personnel intensive nature of the manual war game without incurring the added cost of acquiring and fielding a new micro-computer system.

The central problem addressed in this study is how to provide a tool for training company commanders and platoon leaders, i.e. junior leaders, in MOUT. This tool, in the form of a micro-computer driven, training, war game, must be evaluated as to the increased training it provides over existing techniques and at what costs. Additionally, the validity of such a war game as a model must be investigated.

Research Objectives

This thesis has three specific research objectives. First, the BLOCKBUSTER manual training war game is used as a basis to develop a methodology which utilizes the graphic, computational, and storage capabilities of a micro-computer. This will enhance the realism, decrease the playing time, and reduce the personnel requirements for playing BLOCKBUSTER. Second, a basis for evaluating the potential performance of such a war game as compared to existing models is developed. Third, the validity of such a war game as a model or urban combat is analyzed and presented.

CHAPTER II

RESEARCH APPROACH

In this chapter, some of the developmental and operational war games which are currently in use by the U.S. Army are examined. The BLOCKBUSTER manual training war game will be examined since this war game is currently being used to train junior leaders in MOUT. An examination of BLOCKBUSTER will help define the general attributes desired in a micro-computer driven war game. The ACABUG and ARTBASS computer based war games will be examined to highlight techniques which will be applicable to the developments of a micro-computer driven urban war game. The capabilities of the MICROFIX micro-computer system will also be examined as a potential host for this urban war game. A summary of those features which can be adapted to the micro-computer driven war game and an outline of those areas which are not adequately modeled by the existing systems will also be presented.

BLOCKBUSTER

BLOCKBUSTER is a manual training war game designed for conducting leader training in MOUT. Players move and employ miniature vehicles and dismounted unit counters on an iconic, three dimensional scaled terrain board representing a typical Western European village and surrounding terrain. BLOCKBUSTER is the only manual training war game currently in use by the U.S. Army which allows players to fight "in" and "from" individual buildings. It is currently being used at officer basic and advanced schools as well as in some units to train company commanders and platoon leaders.

Target Audience

The target audience is primarily the company commander and platoon leader, although the system can also be used to train squad leaders.

Level of Units Represented

Players can "move" and "fire" individual vehicles and dismounted infantry squads or fire teams. Combat support units; i.e., field artillery pieces, fixed and rotary wing aircraft, air defense weapons, etc., are available for the use within the limits of the scenario. Provisions are also made in the rules to address combat service support within units, i.e., ammunition resupply, evacuation of casualities, etc.

Level of Representation of Urban Terrain

Players can "enter", "fight from", and "fight in" individual buildings. Individual buildings can be rubbled to create obstacles or clear fields of fire. A village is represented using iconic models of individual buildings represented at a scale of one inch for seven meters. A village layout can be modified from scenario to scenario and is overlayed on the larger terrain board which represents the surrounding terrain at a scale of one inch for fifty meters.

Role of the Player

The role of the individual player is dependent on the scenario and the number of personnel playing the war game. Players must move individual counters, determine line of sight and range to target, and decide which engagements will occur. An umpire determines the results of each engagement in a one-on-one examination based on firer and target types, range to target, and current status of the firer and target.

Playing Time/Personnel Required

Playing time varies greatly depending on scenario complexity, size of engagement, and familiarity of players and umpires with the war game system. A minimum of four players and two umpires are required. A six to eight hour BLOCKBUSTER exercise simulates fifteen to thirty minutes of actual combat.

Limitations

The dual scale terrain board creates confusion in line of sight and range determination. The manual nature of the game often results in a tradeoff of the degree of realism portrayed in the model for a reduction in playing time and personnel requirements.

A detailed description of the BLOCKBUSTER war game is outlined in Reference [7]. Appendix A outlines a typical training scenario which could be played using the BLOCKBUSTER war game system.

Desired Attributes

L. Robert Ogus [22] provides a list of the advantages of war game based training (Table 2); its limitations (Table 3); and the uses of the computer in war game based training (Table 4). BLOCKBUSTER is an existing manual training war game which we are examining as our basis for a computer-driven training war game. Using this information,

Table 2. Advantages of War Game Based Training

- OPPORTUNITY FOR EXPERIENCE IN A CONTROLLED ENVIRONMENT.
- LEARNING THROUGH PARTICIPATION.
- POTENTIAL FOR INCREASED MOTIVATION.
- NEAR REAL-TIME FEEDBACK IMPORTANT FOR DECISION-MAKING.
- PHYSICAL AND LOGISTIC SUPPORT ADVANTAGES.
- COMPLEMENTARY TO OTHER FORMS OF TRAINING.

Table 3. Limitations of War Game Based Training

- LESS EFFECTIVE IN TEACHING CERTAIN TYPES OF KNOWLEDGE OR SKILLS THAN OTHERS.
- WARGAMES CAN BE TIME-CONSUMING AND COMPLEX.
- POTENTIAL EXISTS FOR DEVELOPMENT OF UNREALISTIC ATTITUDES.
- WARGAME-BASED TRAINING HAS THE POTENTIAL TO BE MORE EXPENSIVE THAN OTHER APPROACHES TO THE SAME TRAINING PROBLEM.

Table 4. Use of the Computer in War Game Based Training

- TO SPEED UP PLAY.
- TO ALLOW FOR UTILIZATION OF MORE SOPHISTICATED MODELS.
- TO PERMIT STORAGE AND USE OF MORE DATA.
- TO PROVIDE FOR COMPLEX INTERACTION AMONG PLAYERS.
- TO ALLOW FOR EASIER EMPLOYMENT OF STOCHASTIC MODELS.
- FOR EASE OF GATHERING STATISTICAL AND ANALYTICAL PERFORMANCE DATA.
- TO ELIMINATE THE NEED FOR MANY HUMAN PLAYERS OR DECISION-MAKERS.

attributes we desire to incorporate into this new war game system.

Digitized map data is currently available from the Defense Mapping Agency (DMA), with a resolution of twenty-five meters [4]. This allows the modeler to get away from the limitations of the iconic, three dimensional, dual scale, terrain boards used in BLOCKBUSTER. Although, since it is desired that individual buildings be addressable, the iconic representation of buildings used by BLOCKBUSTER should be adapted to preclude the need to develop a new data base which represents individual buildings.

The display of terrain and the players interaction with this display must attempt to make up for the lack of an easily interpretable three dimensional model. The player must be able to analyze the terrain on which he is fighting prior to the start of the war game and during play. In the area of terrain appreciation the following elements are desired attributes.

Easy Recognition

The player must be able to readily interpret information provided by the display.

Overall Area of Operations

Players should be able to see how their mission fits into the overall plan of the battle, i.e., where adjacent units are located and what are their missions.

Specific Area of Operations

The player must be able to zoom in on the battlefield he is expected to fight upon.

Urban Areas of Operations

A detailed village map at a smaller scale than those found in other areas of operations displays should be available to provide the player with more information.

Intervisibility Plots

These plots would allow players to determine whether or not line of sight exists from any point to any other points. These plots would also allow players to obtain an intervisibility plot for a player specified cone of fire.

Perspective Viewing

This type of viewing would allow players to call up a three dimensional perspective view from certain pre-designated locations.

The computer is also to be utilized to provide a more realistic representation of urban combat. The following attributes are desired to allow the player a realistic amount of control of the battle.

Level of Units Represented

As a company commander or platoon leader, the player must be able to address individual vehicles and dismounted infantry squads or fire teams, although his control over these units will vary.

Placement of Friendly Forces

The player must be allowed to determine the initial locations of his subordinates, based on his assessment of the mission, and the terrain on which he is to operate.

Direct Movement

The player must be allowed to direct when and where his subordinates will move and define what route they will use.

Define Engagement Priorities

For each addressable subordinate, the player must be able to dictate the priority in which this subordinate will engage targets as well as define the area in which this unit will engage targets.

Allocate Combat Support and Combat Service Support Assets

Within the limitations of the scenario being played, the player will have assets available for his use in tasks such as indirect fire support, obstacle creation, ammunition supply, and close air support. These assets must be included in the model for the player's use.

We then increase the realism and reduce the playing time by allowing the computer to control the following attributes.

Disseminate Intelligence Information

The computer makes all line of sight determinations based on the digitized terrain data. The computer displays only those enemy units which are visible to friendly forces.

Control Engagements

The computer, based on line of sight and the player's engagement priorities, determines which firers engage which targets without further player input.

Control Movement

Based on player defined routes and digitized terrain data, the computer "moves" units and determines their locations when necessary.

Control Combat Support and Combat Service Support Resources

The computer controls the availability of combat support and combat service support resources. The computer will also determine the impact point of indirect fire and close air support.

ACABUG

The American Canadian Australian and British Urban Game (ACABUG) is a force-on-force, high resolution, computer assisted war game designed to "improve the capabilities of the four nations for undertaking analytic studies of MOUT related issues, whether they be weapons design, force mix, or tactical problems" [2]. As an analytic tool, ACABUG attempts to model urban combat in as much detail as possible.

ACABUG is a discrete event simulation which requires its players to move units on a three dimensional terrain board, and establish line of sight, detection and engagement opportunities. The role of the computer is to assess the results of the engagements, feed information back to the players, and perform various bookkeeping functions. Although it is designed for analytical purposes, it could be adapted for training purposes. ACABUG is the only force-on-force, high resolution war game which addresses MOUT issues. Figure 3 diagrams the logic flow during an exercise.

Target Audience

If adapted as a training war game, the high resolution present would allow it to train company commanders, platoon leaders, and squad leaders.

Level of Units Represented

Units are represented at the same level as those in BLOCKBUSTER.

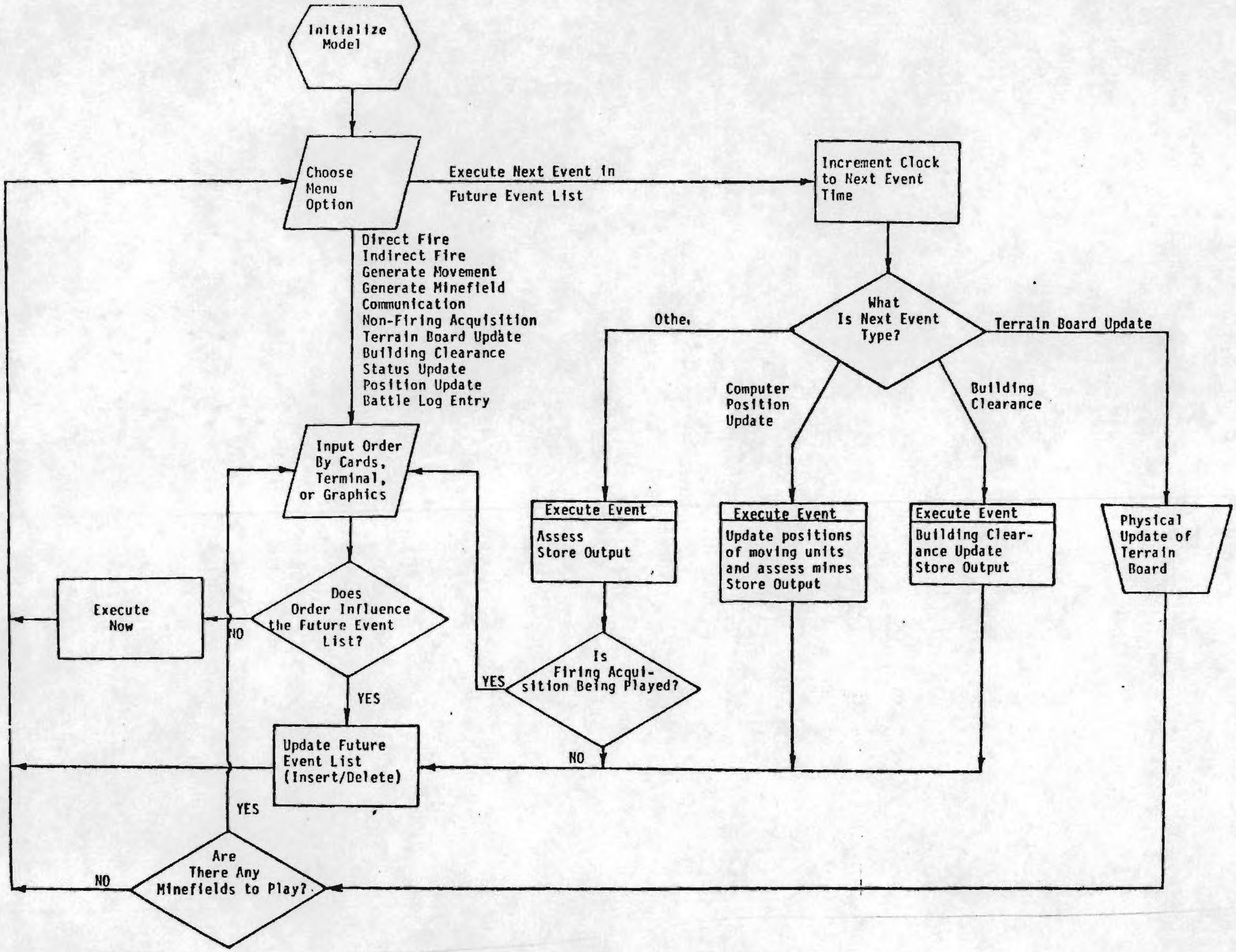


Figure 3. ACABUG Exercise Logic Flow

Level of Representation of Urban Terrain

The computer model of buildings allows the player to place his units within specific rooms of individual buildings. Direct or indirect fire against a building is resolved into the effect of such fire against the building itself and the units currently occupying the building. When players desire to assault and clear a building of opposing forces, they need input only the building identification and the number of attacking and defending forces rather than moving forces from room to room in the building.

Role of the Player

Similar to BLOCKBUSTER; however, the computer replaces the umpire in resolving engagements. Additionally, movement rates and new positions can be computed by the computer rather than measured by the player.

Playing Time/Personnel Required

Although ACABUG is still in development the final goal of the program is to simulate ten to fifteen minutes of actual combat in eight hours of playing time. Twelve to sixteen people are required for an exercise.

Limitations

ACABUG retains the iconic, three dimensional, terrain board for movement and line of sight determination although the larger board uses only a single scale. The result is an increase in the amount of intelligence information available to individual players. The ACABUG system utilizes a PERQ micro-computer with 1 megabyte of RAM and

24 megabytes of hard disk memory to assist in the play of the game. The graphics capabilities of this system are used to speed data input. Digitized terrain data is used for long range line of sight determination only. Still time and personnel intensive, it presents a greater amount of detail than is currently present in BLOCKBUSTER.

The detailed nature of the ACABUG war game precludes the easy adaptation of this system to a smaller micro-computer. However, some features and algorithms can be used to enhance the realism of the war game system we wish to develop.

Building Clearance Algorithms

Building types portrayed in BLOCKBUSTER are consistent with those used in ACABUG. Inputs required by this algorithm are directly available in our model.

Effects of Direct and Indirect Fire on Buildings

Analytical results obtained by ACABUG could be used to degrade the defensive values of buildings as a result of direct and indirect fire without using the complex algorithms required by ACABUG.

Suppression Algorithms

Directly applicable to BLOCKBUSTER, although not currently validated; (Page 30 and Appendix A of Reference [2]).

Additional descriptions of ACABUG are available in [3], [1], and [2].

ARTBASS

The Army Training Battle Simulation System (ARTBASS), a computer driven training war game, is an outgrowth of experience gained through several years of using the Combined Arms Tactical Training Simulation (CATTS), a computer assisted training war game. The model is based on terrain data digitized at a 25 meter resolution. It is capable of modeling a total of 200 units during a simulated exercise. ARTBASS was developed to train battalion commanders and their staffs.

ARTBASS is a continuous-time simulation allowing players to input their orders at any time, but conducting the simulation in one minute time steps.

ARTBASS uses the speed and computing power of a dedicated computer and the information provided by the digitized data to generate terrain maps at five different scales with a maximum redraw time of ten seconds. Additionally, the color graphics displays can generate perspective views, display control measures such as unit boundaries and phase lines, and generate a field of fire fan which displays intervisibility based on observer position and height and target height. The exercise runs in near real time.

Target Audience

The target audience is primarily the battalion commander and his staff, although brigade commanders can also be trained.

Level of Units Represented

Units are represented primarily at the platoon level, although some units are represented at the squad level.

Leve of Representation of Urban Terrain

Urban terrain is addressed as a terrain type; that is individual buildings are grouped together and classified as urban terrain in much the same way that individual trees are grouped together and classified as wooded terrain.

Role of the Player

The game is designed to simulate a combat environment as closely as possible. Players are not required to interface directly with the computer. Orders are input by players over simulated communication channels.

Playing Time/Personnel Required

Play is near real time, the game requires eight people to operate in addition to players being trained.

Limitations

The level at which units are portrayed and the lack of an adequate representation of urban terrain limit the use of this war game as a tool for training junio leaders in MOUT. Additionally, the size of the hardware required to drive this system is prohibitive (see Figure 4 for typical layout).

The use of color graphics and digitized terrain data by ARTBASS provide algorithms which can be adapted for use by this new war game system.

Digitized Terrain Data

Digitized terrain data at a resolution of 25 meters is currently available for Europe, Northeast Asia, Southwest Asia, and the Middle East. Each 25 meter data point is currently stored as two 32 bit words. It is possible to preprocess this data to reduce storage requirements at a cost of reducing the amount of information available

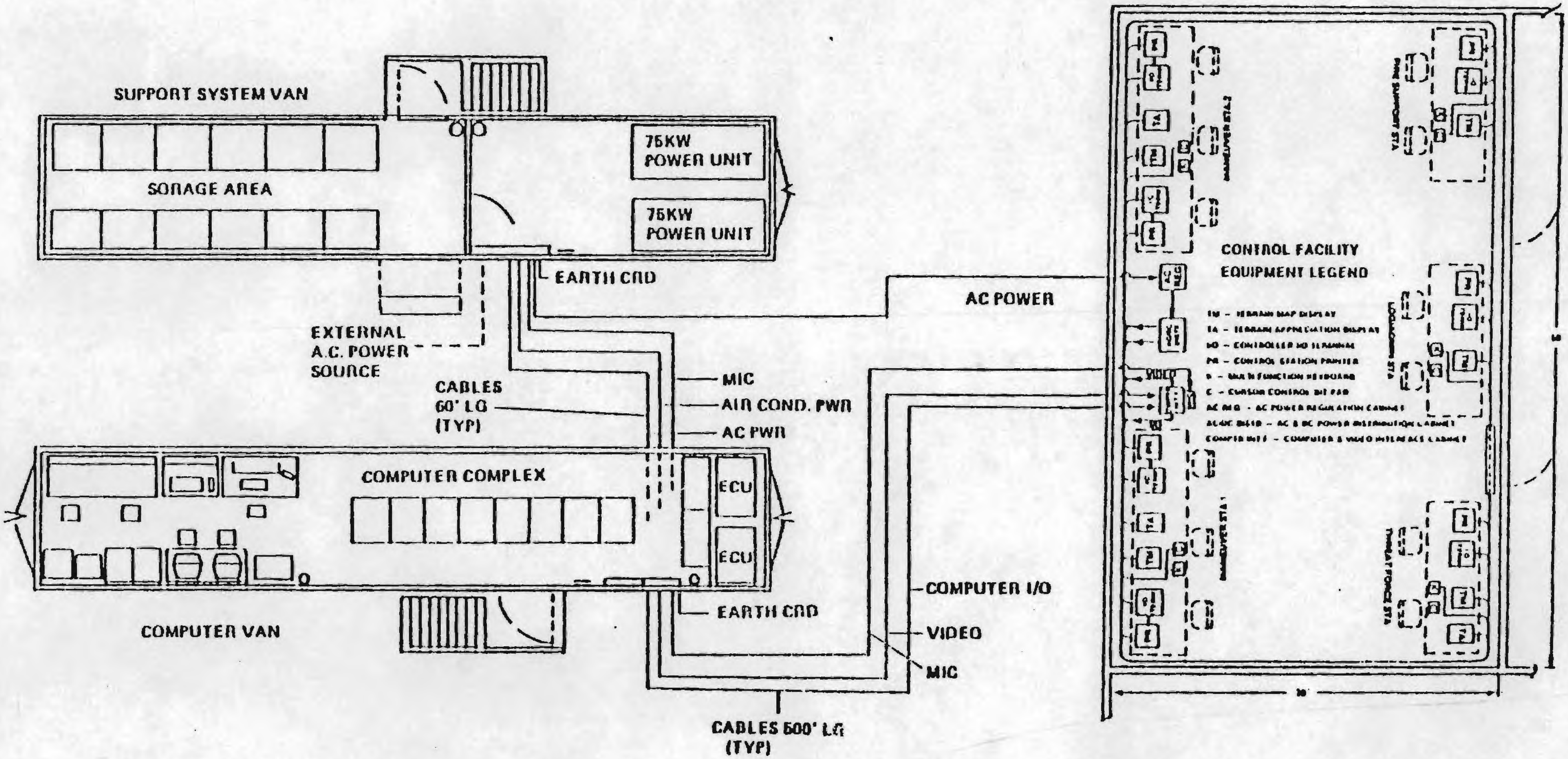


Figure 4. ARTBASS Facilities Layout

from each data point. Additionally, ARTBASS provides algorithms for using this data for line of sight determination and movement rate determination which are readily adaptable for use on a micro-computer.

Terrain Appreciation

Algorithms are available for using digitized terrain data for display of line of sight fans and three dimensional perspective viewing.

Defining Engagement Priorities

ARTBASS does not require that the players define which engagements will take place thereby limiting the amount of control the player has over directing the actions of subordinate units. Algorithms exist for determining which units are engaged based on proximity, threat to firer, and type of target.

Movement of Units

Algorithms exist which allow the player to direct a unit to move along a player specified route. The computer then "moves" the unit along that route, modifying movement rates as appropriate, and displays current positions when the display screen is updated.

Portrayal of Combat Support and Combat Service Support

All combat support and combat service support assets available to a front line unit are modeled by ARTBASS.

Additional information on ARTBASS is available in [11] and [4].

MICROFIX

MICROFIX is a stand-alone desk top micro-computer system consisting of a microprocessor and up to ten inter-connected peripheral subsystems.

Although this may not be the system for which a war game such as that proposed here would be used, it represents the capabilities of a micro-computer system which has already been fielded and typical limitations which should be addressed when considering a micro-computer based war game. Sixty-eight of these systems have currently been fielded with an additional 172 to be fielded in the near future.

Figure 5 shows a fully configured MICROFIX micro-computer system.

The central processing unit is an Apple II+ micro-computer, using a standard 8 bit word, with the following significant enhancements:

- 128 K RAM Card -- for increased internal memory
- CPS Interface Card -- to control printer and provide time keeping capability
- Z80 Microprocessor Card -- doubles the clock speed of the CPU and enables the use of the CP/M operating system
- Disk Controller Card -- operates two 5 1/4" diskette drives
- Video/Microcomputer Interface Card -- operates a video disk player

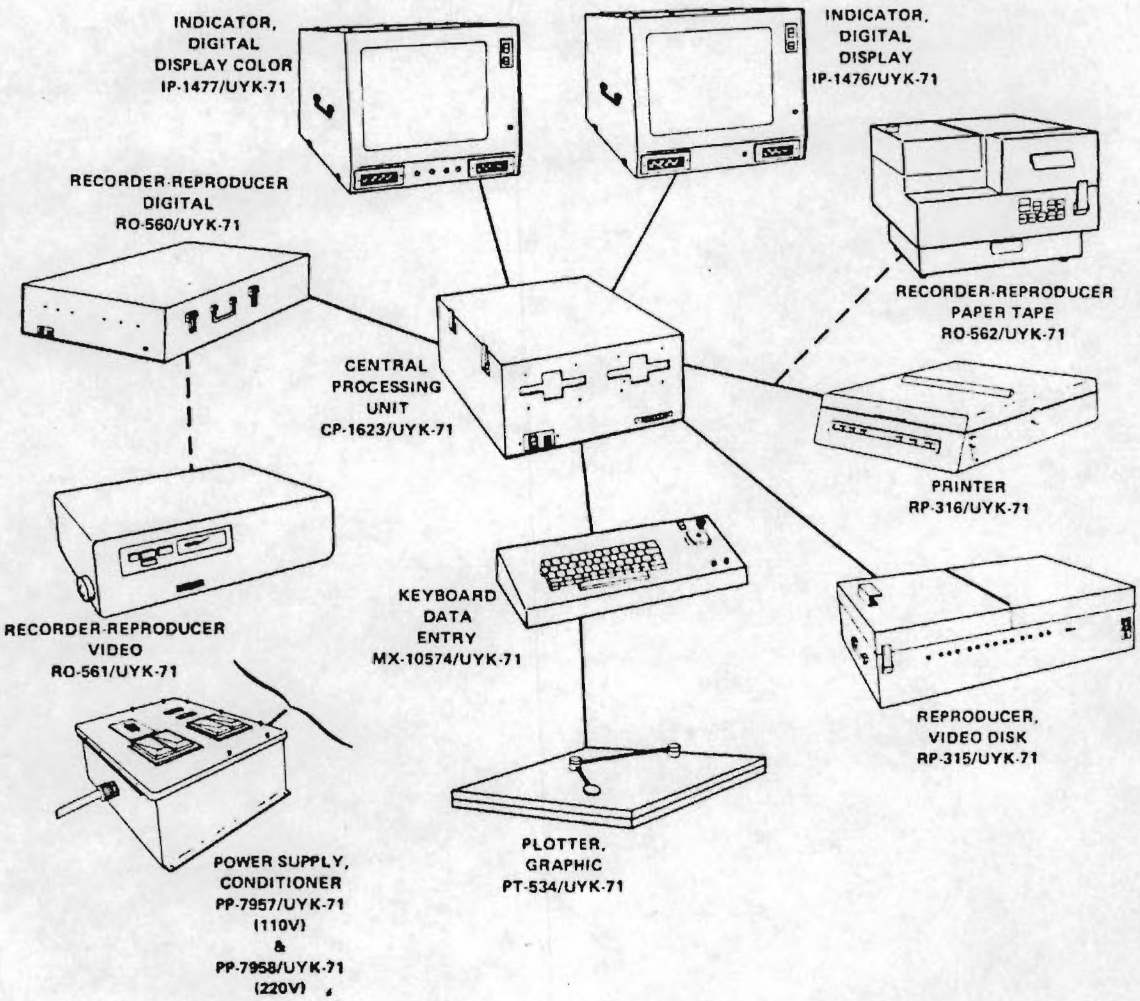


Figure 5. MICROFIX Micro-Computer System Configuration

- Video/Microcomputer Interface Card -- operates a video disk player
- Transporter Interface Card -- provides hard disk capability
- VB-3 and VB-4 Video Cards -- provides graphics capability

Major Peripheral subsystems include:

- Monochrone Monitor -- for display of text
- Color Monitor -- displays color graphics in 260 x 200 resolution
- Keyboard Assembly -- detached; provides full 128 ASCII character set and joystick
- Hard Disk Drive Assembly -- provides 20 Mbyte hard disk storage capability
- Graphics Tablet -- allows user to draw sketches on graphics screen
- Video Disk Player -- stores and displays standard military map sheets of various scales

Software included with system provides a data base management system (dbms) for storage, manipulation and display of text and graphics information. The dbms is currently used to file and display military intelligence information. This information includes current unit position, a "history" of the last ten locations of a unit, textual information as to the units strength, equipment, status, etc. Data sorting is also provided for information on units based on user specified parameters such as activity, type of unit, and units within a designated area [19].

Summary

A review of current micro-computer and war gaming technology reveals much which can be adapted for use in a micro-computer driven training war game.

MICROFIX

- fielded, multi-use micro-computer system
- video disc map display with zoom and pan features
- unit positional display based on universal trans mercator (UTM) coordinates (military standard)
- file input and file sorting for unit information
- color graphics capabilities for display and input

ACABUG

- analytic urban combat data
- building clearance and suppression algorithms
- detailed data files which can be selectively added as needed and as memory allows

ARTBASS

- digitized map data at a 25 meter resolution
- terrain appreciation algorithms
- line of sight, movement and engagement selection algorithms

These existing methods can be selectively added to a micro-computer driven training war game based on the capabilities of the system for which this war game is designed and consistent with the degree of realism needed to train junior leaders in MOUT tactics. There remain areas which are not adequately modeled by existing systems. These areas are discussed briefly below, and in detail in Chapter III.

Representation of Urban Terrain

Using the digitized map data and data handling techniques available from ARTBASS would eliminate the requirement for the iconic, three dimensional, scale terrain boards used by BLOCKBUSTER and ACABUG. A resolution of only twenty-five meters does not result in an adequate portrayal of individual buildings. A methodology for providing an adequate representation of urban terrain will be discussed.

Limiting Battlefield Intelligence

ARTBASS displays only those enemy units which can be "seen" by friendly units. The methodology used assumes a 360 degree field of view. The lower level at which units are to be represented suggests that this methodology would be unrealistic and should be modified.

Model Dynamics

The methodology for controlling subordinate units in ARTBASS will be used as a starting point. The lower level at which units are represented and the absence of a requirement to provide real time output necessitates some changes in this methodology.

CHAPTER III

MAJOR METHODOLOGICAL ISSUES

General Assumptions

It is important at this point to address some general assumptions which apply throughout this discussion of methodological issues.

1. The methodologies described here can be applied to both friendly and enemy forces without significantly enhancing or degrading the capabilities of either side.
2. The (x,y,z) coordinates of any point are strictly positive.
3. Buildings can be grouped into four basic building types for the purposes of resolving direct and indirect fire against units occupying these building types.
4. Structural modifications and the rubbling of buildings can also be aggregated by building type.
5. Buildings can be represented as rectangular parallelepipeds of known dimensions for the purpose of determining line of sight.
6. The elevation of a building has been pre-determined and stored in the array which defines the building.

7. Each unit represented concentrates its primary detector/acquisition assets in a player specified sector of responsibility. Only enemy units which have moved or fired and are within the sector of responsibility will become candidates for detection/acquisition and be checked for existence of line of sight.

8. In urban terrain, it is assumed that LOS blockage is most likely to result from buildings, therefore, LOS blockage is checked first against buildings before checking against base elevation and vegetation.

9. Units will not enter buildings unless instructed to do so by the player. Units moving through urban terrain may be displayed as occuyping a building, due to the manner in which movement is conducted, without actually occupying a building. A reduced movement factor for movement within urban terrain should account for units moving around rather than through buildings. If, at the time line of sight is computed, a unit occupies the same (x,y) coordinates as a building, it is assumed that the unit will have the same elevation as the underlying terrain and line of sight to this unit will be blocked from all directions except for line of sight which is traced from a potential observer through this potential target's sector of responsibility.

10. LOS to a potential target which occupies a building is possible only if the observer is within the sector of responsibility of the potential target; i.e., the target is facing the observer.
11. An observer will only be allowed to attempt to detect/acquire a target which: (a) has moved or fired in this current time step; (b) is within the sector of responsibility of the observer; (c) has an uninterrupted line of sight, and (d) is not currently detected/acquired by this observer.
12. Neither side has detected/acquired any targets at the start of the war game.

Urban Terrain

The need to represent individual buildings in this war game and the limited memory capabilities of a micro-computer would see to conflict when providing an adequate representation of urban terrain. It is desirable to utilize the digitized terrain data available from the ARTBASS war game system because of the amount of information encoded into the two 32 bit words which define a 25 square meter grid block and the availability of this digitized data for a variety of areas through the Defense Mapping Agency. Additionally, increasing the resolution of the digitized terrain data is likely to make the memory requirements unwieldly for a micro-computer.

The U.S. Army in FM 90-10, "Military Operations on Urban Terrain (MOUT)" [20] lists nine different building types commonly found in Western Europe. BLOCKBUSTER divides these nine building types into four groups based on the effects of direct and indirect fire on these building types and uses a standard casting mold to construct the three dimensional models of these buildings. The BLOCKBUSTER representation suggests that buildings can be represented as graphics primitives and overlayed on the digitized terrain data base.

Building the Urban Overlay

The grouping of buildings into four basic structural types provides a graphics primitive for the modeller to use when preprocessing the terrain data during game development. The initial iconic representations can be developed for each of the four structural types as well as the rubble produced by each of the structural types. These primitives can then be instanced using map coordinates, an orientation azimuth, and a scale factor to develop the city overlay and city map [14]. Once the city map has been developed, the elevation of the corner points of each building must be determined.

Elevation determination is made using an algorithm developed for use in the U.S. Army's Combined Arms Tactical Training Simulation (CATTS) [11]. This algorithm is also used to determine elevation of a unit after movement and during line of sight checks. Implicit in this algorithm is the assumption that the elevation at any point within a terrain block is determined based on a linear interpolation of the elevations at the four corner points of the terrain block. By further assuming that the elevation of a structure is determined to be the elevation of the lower left corner point plus the standard height determined by the building type, we provide a method for defining the elevation of building during the preprocessing stage of development.

A brief description of the ELEV subroutine follows (the program listing is found in Appendix B):

1. Entry into the subroutine brings the (x,y) coordinates of the desired point.
2. These (x,y) coordinates are converted to determine the address of the terrain block they occupy (defined by the coordinates of the lower left corner point of the terrain block) and elevations of the four corner points are drawm from memory.
3. The elevation is then computed based on the x,y coordinates of the desired point, the coordinates of the terrain square they occupy and the known elevations of the terrain square's corner points.
4. Subroutine ELEV returns the elevation of the desired point.

Math Model. (See Figure 6). Given:

y, YC, YCT, x, XC, Z1, Z2, Z3 and Z4, all strictly positive

By holding YC constant, the elevation at any point, x, along the line segment $\overline{AB}$ becomes:

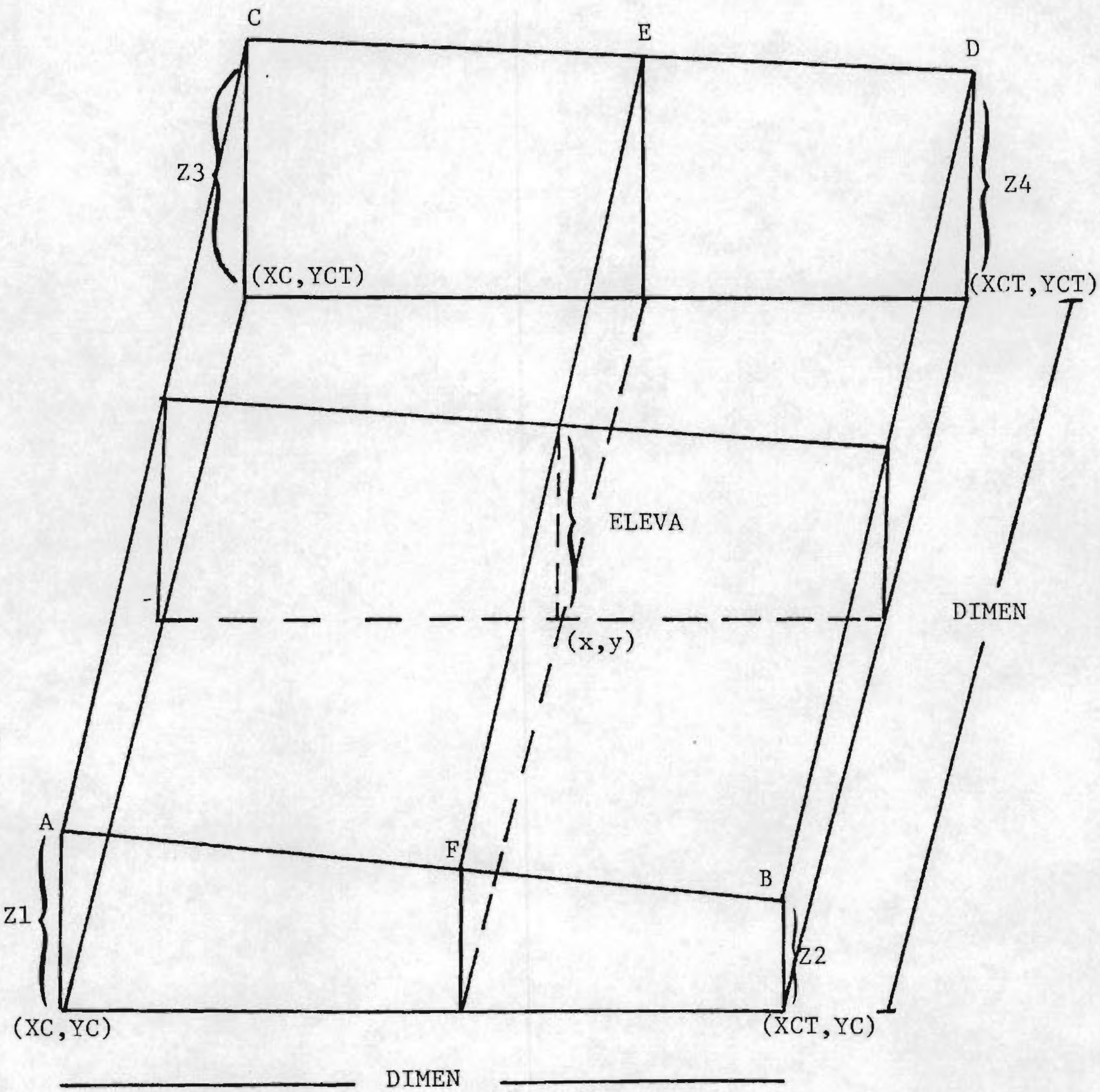


Figure 6. Elevation Determination

$$Z_{YC}(x) = Z1 + \frac{x-XC}{XCT-XC}(Z2-Z1). \tag{1}$$

Similarly, by holding YCT constant, the elevation at any point, x, along the line segment $\overline{CD}$ becomes:

$$Z_{YCT}(x) = Z3 + \frac{x-XC}{XCT-XC}(Z4-Z3) \tag{2}$$

Having solved for $Z_{YC}(x)$ and $Z_{YCT}(x)$ by holding x constant we can now solve for the elevation at any point (y) along the line segment $\overline{EF}$:

$$ELEVA = Z_{YC}(x) + \frac{y-YC}{YCT-YC}(Z_{YCT}(x)-Z_{YC}(x)). \tag{3}$$

Since the terrain block is a square:

$$DIMEN = XCT - XC = YCT - YC$$

Substituting Equations (1) and (2) into Equation (3) and simplifying we obtain:

$$ELEVA = (A1_{xy} + B2_{x} + C3_{y} + D4)/DIMEN^2 \tag{4}$$

where:

$$A1 = Z1 - Z2 - Z3 + Z4$$

$$B2 = YCT(Z2-Z1) + YC(Z3-Z4)$$

$$C3 = XCT(Z3-Z1) + XC(Z2-Z4)$$

$$D4 = XCT((YCT \cdot Z1) - (YC \cdot Z3)) + XC((YC \cdot Z4) - (YCT \cdot Z2)).$$

Demonstration. Appendix B provides a sample run of this program which determines the elevation of the 4 corner points of a building. The algorithm computes the elevation of these corner points in approximately one second. This program was derived from the mathematical model of a similar subroutine outlined in [11].

Urban Terrain Grid

Building location (i.e., the (x,y) coordinates of each corner points), elevation, and type are then used to construct a file of buildings.

An Urban Terrain Grid is then constructed. The purpose of the Urban Terrain Grid is to reduce the number of checks for the intersection of two line segments which must be made. Since terrain data is divided into square blocks of known dimension [11], the Urban Terrain Grid should fit within these terrain blocks so that when a terrain block containing urban terrain is to be checked for line of sight blockage it will be checked for blockage by urban terrain before being checked for blockage by ground elevations. It is recommended that the dimensions of the urban terrain blocks be

50 meters by 50 meters. Such a large block would reduce the number of buildings occupying more than one block and at the same time would evenly divide a 1km by 1km terrain block. Once the grid has been established, a list of what buildings occupy (wholly or partially) which grids is developed and stored in two linked lists (see Figure 7). In this manner only buildings which are within, at most, 70 meters of the observer-target line will be checked for intersection with the observer-target line and possible line of sight blockage.

BOXES

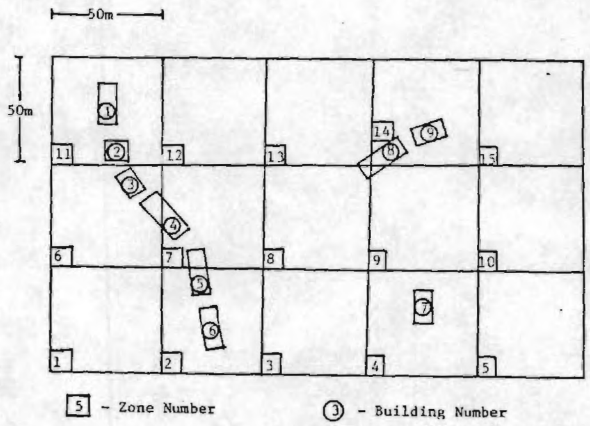
The BOXES subroutine is used to determine which Urban Terrain Grids are intersected by the observer target line. This subroutine is also adapted from the U.S. Army's CATTS models [11]. This subroutine can also be used to determine which digitized data terrain grid squares will be called into memory for line of sight determinations and movement.

A brief description of the algorithm follows (the program listing of the subroutine is found in Appendix C):

1. Entry into the BOXES subroutine provides the (x,y) coordinates of the observer-target pair.
2. Based on the observer target coordinates, and the dimensions of the urban terrain grid, BOXES computes the normalized (x,y) coordinates of the observer and target urban terrain grids.
3. The algorithm then determines the number of grid blocks crossed by the observer-target line (O-T

Zone	Building Number
1	0
2	5,6
3	0
4	7
5	0
6	3,4
7	4,5
8	8
9	8
10	0
11	1,2
12	0
13	8
14	8,9
15	0

Step 2



Step 1

Building Number Array BNA(x)	x=	Zone-Bldg # Key ZBK(x)
5	1	1
5	2	1
7	3	3
3	4	3
4	5	4
4	6	4
5	7	6
8	8	8
8	9	9
1	10	10
2	11	10
8	12	12
8	13	12
9	14	13
0	15	15
0	16	15

Step 3

To find Bldg. # to check for LOS blockage

1. Determine the ith grid zone crossed. Let

   n_i = ith zone crossed by the O-T line

2. Determine values of zone-bldg # key (ZBK(x)) in position n_i and n_i+1. Let

   $A = ZBK(n_i)$

   $B = ZBK(n_i+1)$

3. Determine the number or buildings in zone n_i. Let

   $C_i = B-A$

   if $C_i = 0$ - no building in this zone. Go to Step 1.

4. Build array of buildings to be checked for LOS blockage (AB(x)). For

   $j = 1$ to C_i

   $AB(k) = BNA(A+j-1)$

   $k = k+1$

   Next j.

Figure 7. Development of Urban Terrain Grid and the Building in Grid Arrays

line) and the sign of the slope first in the X and then in the Y direction.

4. Based on the sign of the X-slope and the normalized number of grid blocks crossed in the X direction BOXES determines first the normalized X coordinate of the grid line crossed, and then the actual Y coordinate of the intersection of the observer-target line and this grid line. This Y coordinate is then normalized and the normalized (x,y) coordinates of the grid block crossed are stored to be checked for line of sight blockage.
5. Step 4 is repeated for those grid blocks crossed in the Y direction.
6. BOXES then determines the zone numbers crossed by the observer-target line.

Math Model. Let

$$\text{XSTART} = [x_o/\text{DIMEN}] \text{ and } \text{YSTART} = [y_o/\text{DIMEN}]$$

$$\text{YFINISH} = [x_t/\text{DIMEN}] \text{ and } \text{YFINISH} = [y_t/\text{DIMEN}]$$

where (x_o, y_o) are the observer's global coordinates and strictly positive

(x_t, y_t) are the target's global coordinates and strictly positive, and

DIMEN is the dimension of the standard urban terrain block.

The normalized (x,y) coordinates of the observer and target grid blocks are defined to be the coordinates of the lower left hand corner of the grid block.

The coordinates of the other grid blocks are computed using a slope/clipping concept. This concept is used to perform computations in both the X and Y directions, but will be discussed for the X direction only.

To compute the number of grid lines traversed in the X direction

$$\text{X Slope} = |\text{X Finish} - \text{X Start}| \tag{4}$$

The sign of the slope is then determined and the values of two constants are set.

$$\text{SIGNX} = \begin{cases} 1 \text{ if XSLOPE is positive} \\ 0 \text{ if XSLOPE is zero} \\ -1 \text{ if XSLOPE is negative} \end{cases}$$

$$Q = \begin{cases} 1 \text{ if XSLOPE is positive or zero} \\ 0 \text{ if XSLOPE is negative} \end{cases}$$

Q is set equal to zero, if X slope is negative, and then reset equal to 1 after the coordinates of the first grid block crossed is determined.

The X coordinate of the grid line is then transformed into its corresponding global coordinate using Equation (5).

$$XCUR = (X\ START)(DIMEN) \tag{5}$$

The subroutine then proceeds through a loop, once for each grid line traversed as determined by X Slope. The first step of the loop is to determine the global coordinates of the next grid line to be crossed in the X direction, this is computed using Equation (6).

$$XCUR = XCUR + (DIMEN \times SIGN\ X \times Q) \tag{6}$$

It is important to note that, if XSlope is negative, Q is equal to zero only during the first iteration of the loop, afterward Q is set equal to 1.

<u>Example 1</u>: Using line 1, Figure 8 as an example, given

DIMEN = 10

Observer Position = (15,24)

Target Position = (49,41)

Then

(XSTART, YSTART) = (1,2)

(XFINISH, YFINISH) = (4,4)

XSlope = 3, SIGNX = 1, Q = 1

From Equation (5), XCUR = 10 initially and the program would now check for 3 grid blocks crossed in the X direction by the O-T line; using Equation (6) XCUR becomes

$$XCUR = 10 + (10 \times 1 \times 1) = 20$$

so the normalized X coordinate of the first grid block is (20/10) = 2.

Example 2: Using line 2, Figure 8 as an example, given

DIMEN = 10

Observer Position = (51,28)

Target Position = (23,7)

Then

(XSTART, YSTART) = (5,2)

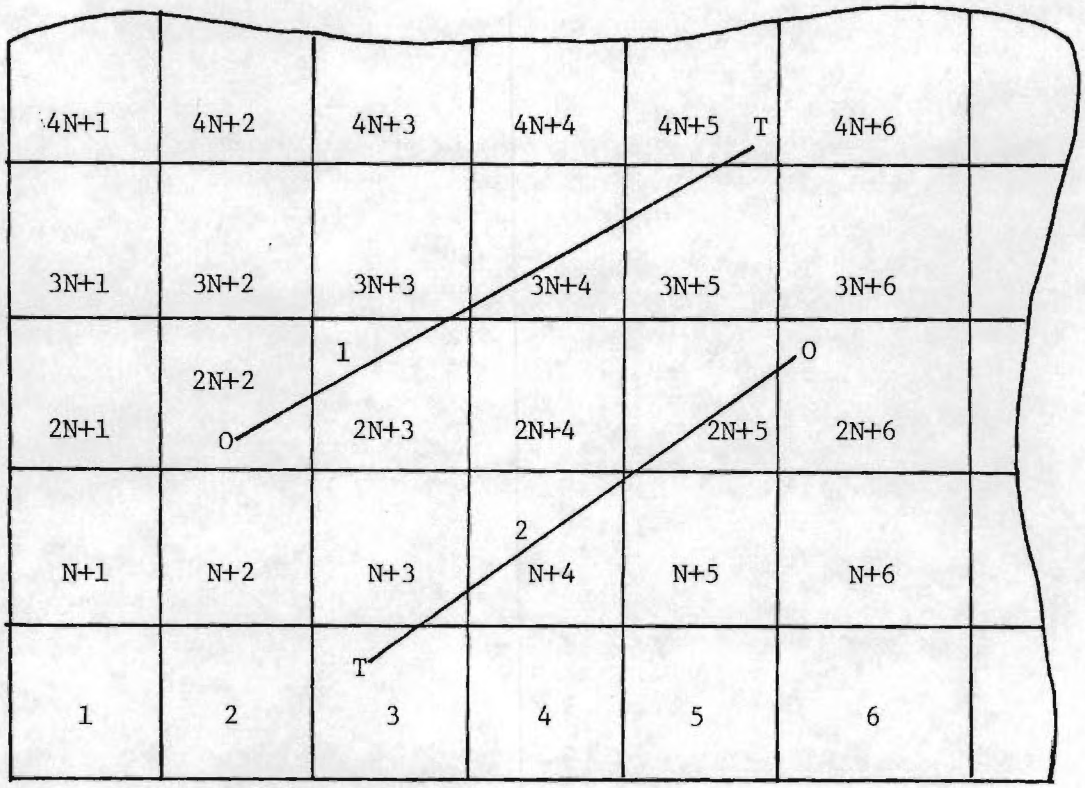
(XFINISH, YFINISH) = (2,0)

XSlope = 3, SIGNX = -1, Q = 0

From Equation (5), XCUR = 50 initially and the program would check for 3 grid blocks crossed in the X-direction by O-T line.

Using Equation (6), XCUR becomes

$$XCUR = 50 + (10 \times -1 \times 0) = 50$$



O - Observer position
T - Target position
N - Number of boxes per row

Figure 8. Example O-T Lines for BOXES

So the normalized X-coordinate for the first grid block is (50/10) - 1 = 4. (Note: on the next iteration of this loop Q = 1).

Once XCUR has been determined in global coordinates XCUR and the global coordinates of the observer and target positions are used to determine the Y-coordinate of the point of intersection of the O-T line and XCUR.

$$Y = \frac{(XCUR(y_t - y_o)) + ((x_t y_o) + (x_o y_t)))}{(x_t - x_o)} \tag{7-A}$$

when given YCUR the X-coordinate of the point of intersection of the O-T line and YCUR is

$$X = \frac{(YCUR(x_t - x_o)) + ((x_t y_o) - (x_o y_t)))}{(y_t - y_o)} \tag{7-B}$$

Equations (7A) and (7B) are from Reference [23]. The normalized Y-coordinate of the grid block is then determined by:

$$Y = [y/DIMEN] \tag{8}$$

<u>Example 1</u> (Continued): Using line 1, Figure (8)

$$Y = \frac{(20 \times (41-24)) + ((49\times24) - (15\times41))}{(49-15)} = 26.5$$

and the normalized Y-coordinate of the urban terrain grid block is

$$[26.5/10] = 2.$$

As a result the urban terrain grid block coordinates of the first grid block crossed in the X-direction are (2,2).

The algorithm then continues until all grid blocks crossed by the O-T line in the X-direction are determined.

The grid blocks crossed by the O-T line in the Y-direction are then determined in a similar manner.

The algorithm then determines the grid block number of the grid blocks identified; this is accomplished using Equation (9).

$$\text{Grid block number} = X + N \cdot Y + 1 \tag{9}$$

where

X = the normalized X-coordinate of the grid block

Y = the normalized Y-coordinate of the grid block

N = the number of grid blocks in a row (constant)

Demonstration. Appendix C provides a sample run of the program which computes the boxes intersected by 5 observer-target lines. The program computers the results in approximately 2 seconds.

CIBLOC

The CIBLOC subroutine determines whether or not the O-T line intersects a given building and, if intersection occurs, provides the x, y and z coordinates of the point of intersection to determine if line of sight is blocked by a building. This algorithm was developed by the author for this methodology and is an application of formulas found in [23].

A brief description of the algorithm follows. (The program listing of the subroutine is found in Appendix D).

1. Entry into CIBLOC provides the (x,y) coordinates of the observer, target, and building corners.
2. Using formulas outlined in Reference [23] for determining the intersection of two line segments CIBLOC sequentially checks for the intersection of the observer-target line and each of the four walls of the building.
3. If no intersections are found the next building is checked.
4. If intersections are found, the x, y, z coordinates of the intersections (at most 2) are stored and used to check for line of sight blockage in the LOSCHECK subroutine.

Math Model.

Given

(x_1,y_1) and (x_2,y_2) - end points of observer target line (strictly positive)

(x_3,y_3) and (x_4,y_4) - end points of building side (strictly positive)

Define P_1, P_2, P_3, P_4 as the homogeneous coordinates of (x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4), respectively.

Example

$$P_1 = (x_1,y_1,1)$$

$$P_2 = (x_2,y_2,1)$$

$$P_3 = (x_3,y_3,1)$$

$$P_4 = (x_4,y_4,1)$$

Further define

$$S_1 = \det(P_1,P_3,P_4)$$

$$S_2 = \det(P_2,P_3,P_4)$$

$$S_3 = \det(P_3,P_1,P_2)$$

$$S_4 = \det(P_4,P_1,P_2)$$

Then if $S_1 \times S_2 < 0$ and $S_3 \times S_4 < 0$ the two line segments intersect. The point of intersection can then be obtained by solving two simultaneous equations in x and y.

$$x(y_1-y_2) + y(x_2-x_1) + (x_1y_2-y_1x_2) = 0$$

and

$$x(y_3-y_4) + y(x_4-x_3) + (x_3y_4-y_3x_4) = 0$$

solving this system of equations yields

$$x = \frac{(x_2-x_1)(x_3y_4-y_3x_4) - (x_4-x_3)(x_1y_2-y_1x_2)}{(y_1-y_2)(x_4-x_3) - (y_3-y_4)(x_2-x_1)}$$

and

$$y = \frac{(y_3-y_4)(x_1y_2-y_1x_2) - (y_2-y_1)(x_3y_4-y_3x_4)}{(y_1-y_2)(x_4-x_3) - (y_3-y_4)(x_2-x_1)}$$

Note that

$$(y_1-y_2)(x_4-x_3) - (y_3-y_4)(x_2-x_1) = 0$$

implies that

$$x_4(y_1-y_2) + y_4(x_2-x_1) = x_3(y_1-y_2) + y_3(x_2-x_1) \tag{10}$$

Since

$$S_3 = x_3(y_1-y_2) + y_3(x_2-x_1) + (x_2y_1-y_2x_1)$$

and

$$S_4 = x_4(y_1-y_2) + y_4(x_2-x_1) + (x_2y_1-y_2x_1)$$

Then equation (10) implies that

$$S_4 = S_3$$

and by similar logic that

$$S_2 = S_1$$

Since intersection occurs only if $S_1 \times S_2 < 0$ and $S_3 \times S_4 < 0$ $(y_1-y_2)(x_4-x_3) - (y_3-y_4)(x_2-x_1) = 0$ will not occur since $S_1 = S_2$ and $S_3 = S_4$ implies that $S_1 \times S_2 \geq 0$ and $S_3 \times S_4 \geq 0$.

$S_1 = S_2 = S_3 = S_4 = 0$ implies that the two line segments lie on the same line. Due to the rectangular representation of the buildings it is not necessary that this line be checked for intersection. If the wall defined by $\overline{P_3P_4}$ is wholly or partially contained by the line segment $\overline{P_1P_2}$ the endpoints which intersect this line segment will be determined when the other walls are checked for intersection.

The points P_1 and P_2 need not be checked for possible line of sight blockage since they are, by definition, the observer and target locations, respectively.

In the unlikely case where the line segment $\overline{P_1P_2}$ is contained in the line segment $\overline{P_3P_4}$, no line of sight check is made, this is equivalent to firing along a wall.

Demonstration. Appendix D provides a sample run of the program which computes the points of intersection of 3 observer target lines with 1 building. The program computes the points of intersection in approximately 3 seconds.

Line of Sight (LOS)

The line of sight (LOS) algorithm described in [11] is adequate for LOS checks in open terrain. The basic mathematical model is described here for the purpose of describing how line of sight checks are conducted when urban terrain is crossed.

Math Model. Given (see Figure 9a):

(x_o,y_o) - observer location

(x_t,y_t) - target location

(x_b,y_b) - location of possible obstruction

all coordinates are strictly positive.

First, compute straight line distance in the zero elevation plane from observer to target, Equation (11), and observer to possible obstruction, Equation (12).

$$D_{o,t} = ((x_t - x_o)^2 + (y_t - y_o)^2)^{1/2} \tag{11}$$

$$D_{o,b} = ((x_b - x_o)^2 + (y_b - x_o)^2)^{1/2} \tag{12}$$

Next, compute the elevation of the O-T line at the point of possible obstruction (Figure 9b). Given

h_t = elevation of target plus target height

h_o = elevation of observer plus height of observer

h_b = elevation of the possible obstruction

To compute h_{LOS}, similar triangles are used.

$$\frac{D_{o,b}}{D_{o,t}} = \frac{(h_{LOS} - h_o)}{(h_t - h_o)} \tag{13}$$

Solving Equation (13) for h_{LOS}:

$$h_{LOS} = h_o + \frac{D_{o,b}}{D_{o,t}} (h_t - h_o)$$

LOS is blocked if $h_{LOS} \leq h_b$ and clear otherwise.

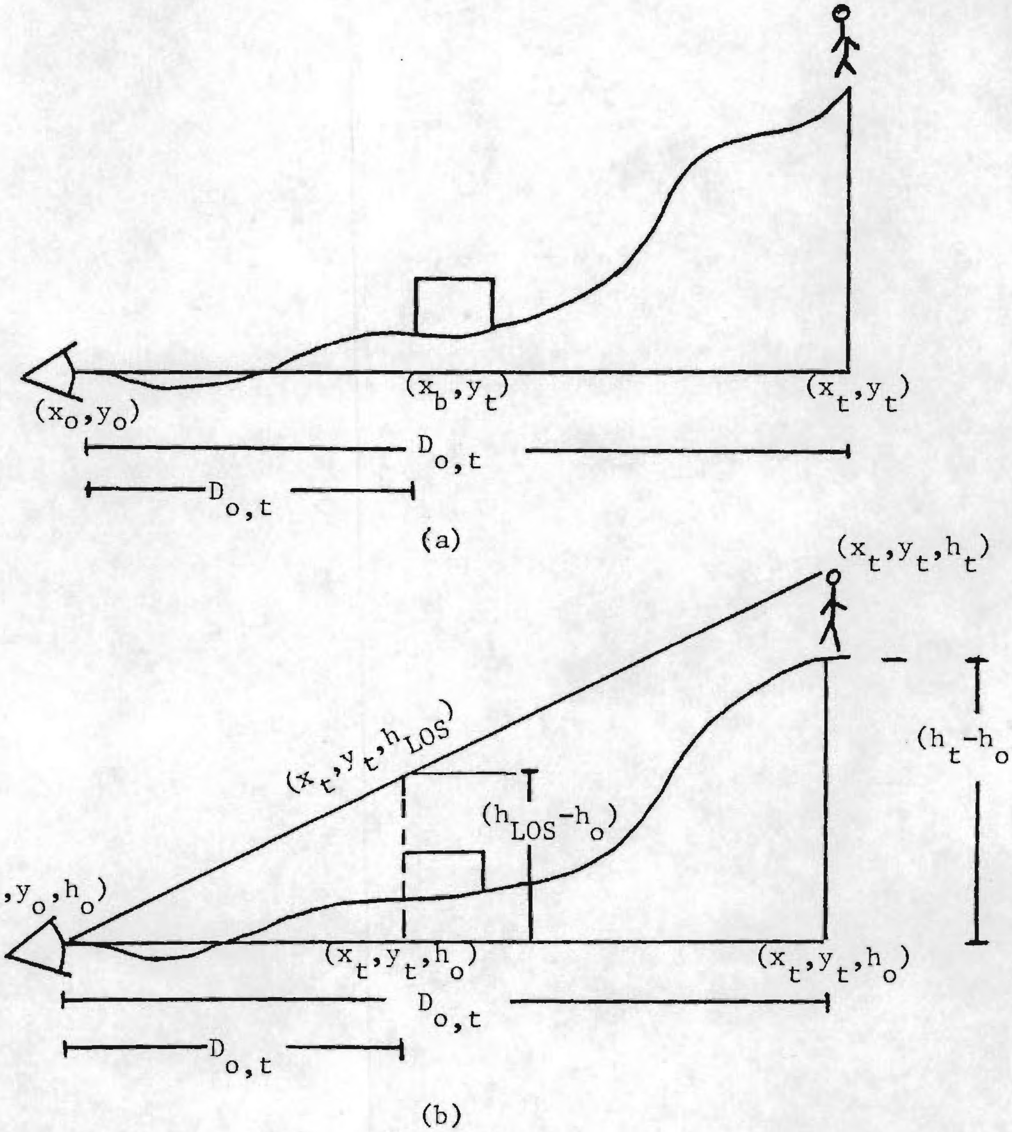


Figure 9. Line of Sight in Urban Terrain

Sectors of Responsibility

Although checking whether or not LOS exists between two units serves to limit the amount of battlefield intelligence available to the players, an additional limitation is provided by the concept of "sectors of responsibility". These sectors of responsibility model the fact that the detection/acquisition efforts of individual weapon systems are not evenly spread in a 360 degree circle around the weapon system. These sectors of responsibility are designated and modified by the player during the course of the game and represent the means by which he controls the fires of his units and the manner in which he develops battlefield intelligence. These sectors of responsibility (SOR) are a logical extension of a common U.S. Army tactic of assigning sectors or fields of fire to insure coverage of the battlefield and to control fires so fire is spread evenly across the front or concentrated on an avenue of approach as the tactical situation dictates. It is felt that applying this constraint to the opposing forces as well as the U.S. forces does not result in an unrealistic constraint (or in enhanced capabilities) on the opposing forces.

SORs are established by the player and represent the primary direction in which a given weapon system will concentrate its search and firing assets. By designating a unit and the left and right boundaries of the SOR the player defines this region.

By using the SOR method the number of LOS checks is reduced since a unit only checks for LOS in its SOR and not to every enemy unit involved in the game. This also limits the amount of information available to the player about the enemy forces involved in the battle especially in the early stages of the battle when very little is known about the enemy forces. Additionally, we can use sectors of responsibility to determine which face of a building a unit can be fired upon through.

The following assumptions are made:

1. Sectors of responsibility can be defined by a sector of radius equal to 5000 meters [11] and a sweep of less than 180 degrees. The assumptions of an SOR sweep less than 180 degrees requires fewer checks than a similar algorithm described in [21].
2. A "self-preservation" zone (SPZ) of radius equal to 33 meters [11] represents the capabilities of secondary detection systems. LOS within the SPZ can only be blocked by a building.
3. A dismounted infantry squad, since it is composed of two fire teams (10-12 men), will be allowed to scan two SORs, all other units/weapon systems will be allowed to scan only one SOR.

See Figure 10 for illustration of SOR and Self Preservation zone.

A brief description of the algorithm developed by the author for this methodology follows (the program listing of the subroutine is found in Appendix E).

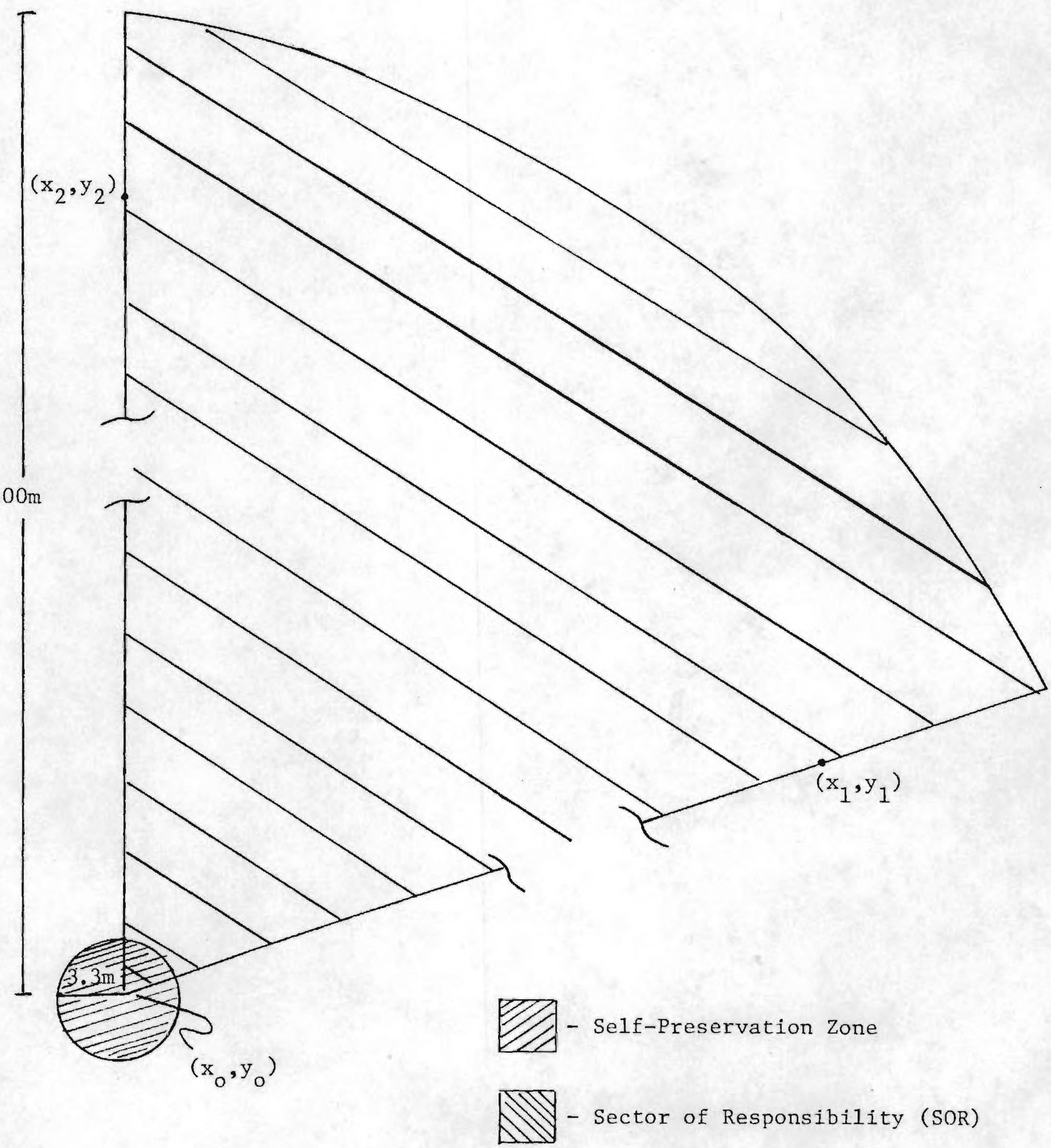


Figure 10. Sector of Responsibility and Self-Preservation Zone

1. Entry into this subroutine provides the (x,y) coordinates of the observers, their left and right SOR boundary points and the targets.

2. Each target is checked against each SOR sequentially to determine which observers may detect/acquire which targets by:

   (a) Checking to see if the target is beyond maximum range (5000 meters), if so, no further checks are required.

   (b) Checking to see if the target is within the units SPZ. If it is within this zone the target is an automatic candidate for detection/acquisition for this observer.

   (c) Checking to see if the target is within the sector defined by the observer's SOR.

3. The subroutine develops a list of which observer target pairs must be checked for LOS blockage before determining detection/acquisition.

<u>Math Model</u>. Given

Observer position = (x_o, y_o)

Target position = (x_t, y_t)

Left and reight boundaries = (x_1, y_1) and (x_2, y_2)

(order is irrelevant)

all coordinates are strictly positive.

The points (x_o,y_o), (x_1,y_1), (x_2,y_2) and (x_t,y_t) can be thought of as vectors about the origin. If we translate (x_o,y_o) to the origin and the other vectors are similarly translated we are able to represent the SOR as a convex cone [5] based at the observer position. The resulting $\begin{pmatrix} x_1-x_o \\ x_1-y_o \end{pmatrix}$ and $\begin{pmatrix} x_2-x_o \\ y_2-y_o \end{pmatrix}$ define the boundaries of a convex cone with its base located at $\begin{pmatrix} x_o \\ y_o \end{pmatrix}$.

Then by the definition of a convex cone the vector $\begin{pmatrix} x_t-x_o \\ y_t-y_o \end{pmatrix}$ is contained in this convex cone if and only if

$$\begin{pmatrix} x_t-x_o \\ y_t-y_0 \end{pmatrix} = \lambda 1 \begin{pmatrix} x_1-x_o \\ y_1-x_o \end{pmatrix} + \lambda 2 \begin{pmatrix} x_2-x_o \\ y_2-y_o \end{pmatrix} \; ; \text{ for all } \lambda i >= 0, \; i = 1,2 \qquad (14)$$

Since the vectors are constants of known value for any given observer-target pair we can solve this system of simultaneous equations for $\lambda 1$ and $\lambda 2$.

Define:

$$a1 = x_1-x_o$$
$$a2 = x_2-x_o$$
$$b1 = y_1-y_o$$
$$b2 = y_2-y_o$$
$$c1 = x_t-x_o$$
$$c2 = y_t-y_o$$

Substituting these into Equation (14) and solving for λ1 and λ2 we obtain

$$\lambda 1 = \frac{(c1b2)+(c2a2)}{(b2a1)-(b1a2)}$$

$$\lambda 2 = \frac{(c2a1)-(c1b1)}{(b2a1)-(b1a2)}$$

Note that (b2a1)-(b1a2) = 0 implies that the left and right boundaries are colinear. Such an SOR is prohibited. To avoid prohibited SORs feedback should be provided to the user which would allow him to determine if his input was correct.

If both λ1 and λ2 are positive, the target is within the observer's SOR. If either is negative, the target is outside the SOR.

Demonstration. Appendix E provides a sample run of this program for 8 observers and 6 targets. The program computes which targets are in which sectors of responsibility in approximately 6 seconds.

Model Dynamics

Combined Continuous-Discrete Simulation

The combined continuous-discrete simulation techniques employed by ARTBASS and the near real time play of the game limit, the control players are able to exert over individual units. Since the computer does not wait for player input to begin the execution of the next time step, delays caused by operating over standard communications networks and the time required to input orders into the computer serve to add realism to the war game. The need to have the exercise run in near real time is not a requirement of this war game. The combined continuous-discrete simulation should, however, be retained.

Math Model. Pritzker [24] describes the three fundamental types of interactions which can occur between discretely changing and continuously changing state variables.

1. Discrete changes may affect the value of continuous variables.
2. Achieving specified conditions for a state variable may cause an event to occur or be scheduled.
3. The functional description of continuous variables may be changed discretely.

In this model of combat, the discrete variables of interest are the number of friendly forces and the number of enemy forces. The continuous variables of interest are the locations of various units. Discrete events such as direct and indirect fire engagements potentially change the value of both the discrete and continuous variables of interest. More importantly, these discrete events only occur if a target is acquired (observed in some manner as to its location and identified as a foe). The modeling of this event is the primary reason the author recommends the use of combined continuous-discrete simulation methods.

The logic flow for the simulation is illustrated in Figure 11. The thirty-second time step between display updates and player input opportunities is based on the thirty seconds of actual combat simulated in one BLOCKBUSTER game turn [7]. This allows for the direct use of many BLOCKBUSTER data tables.

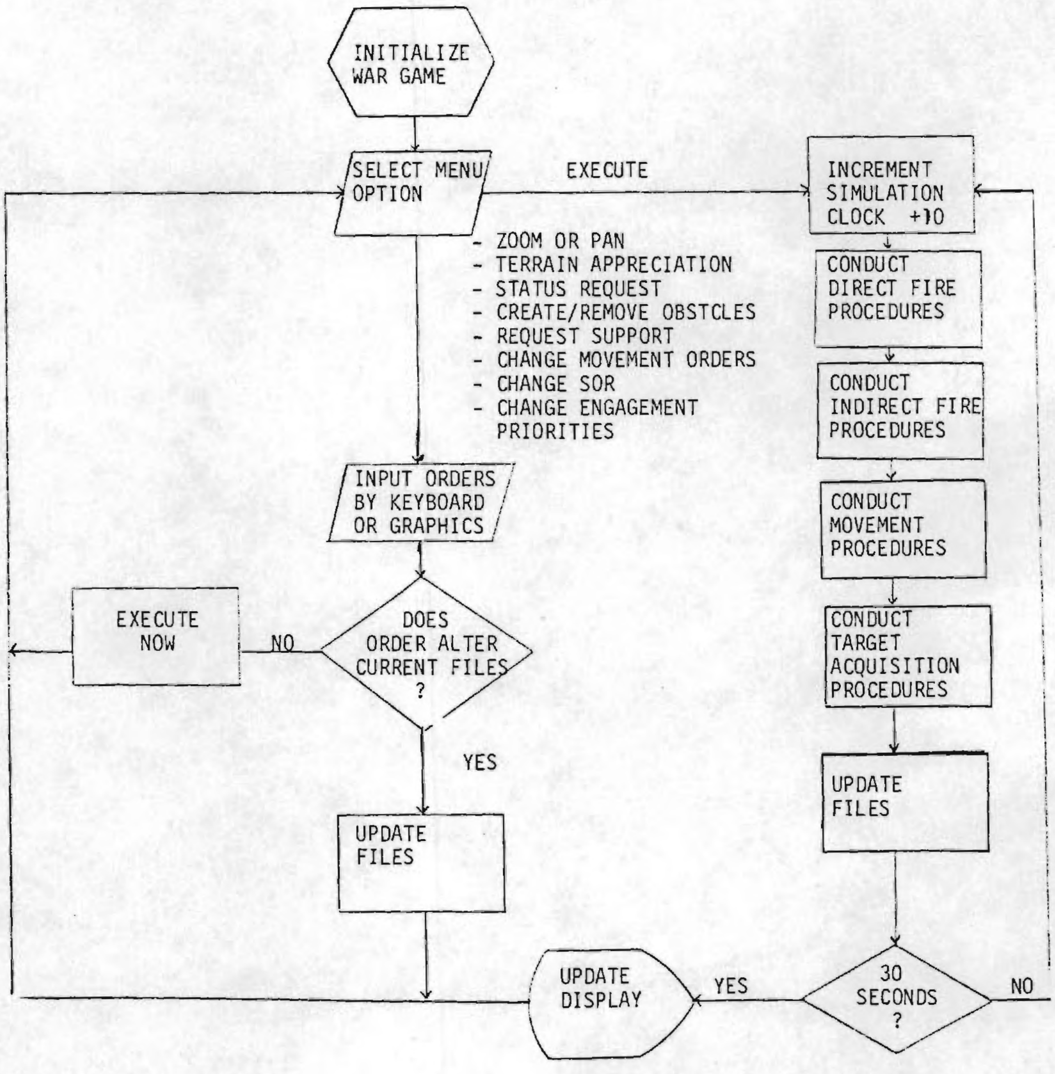


Figure 11. Logic Flow for Combined Continuous-Discrete Simulation

The use of a ten-second time step within the main thirty second loop serves a two-fold purpose. First, it accomodates the high rate of fire of modern weapon systems. The BLOCKBUSTER war game system allows tank main guns to fire three times in one thirty second game turn (Table 5). Second, the ten second time step allows for easier implementation of more sophisticated command and control models while retaining the continuous time stepping techniques as the primary driver for the simulation clock.

Engagement Initiation

ARTBASS resolves direct fire engagements simultaneously. That is, it is assumed that both sides fire at the same time so neither side is attrited until all direct fire engagements have been resolved.

Callahan and Crosby [8] tested the effects of a delayed engagement initiation on Lanchester small unit combat models and found that whomever fires first, and the delay prior to returning fire, could have a significant impact on engagement results. See Figure 12 for their results.

Dr. Donovan Young, in private correspondence with the author, also addresses the issue of who fires first. In a one-on-one duel, assume that each duelist has a probability, p, of obtaining a kill on a given shot. If one duelist fires first, his probability of winning the duel can be shown to be $\frac{p}{1-q^2}$, where $q = 1-p$. The probability of the second duelist winning the duel becomes $q(\frac{1}{1-q^2})$, which is smaller by a factor of q.

Table 5. Threat and U.S. Rate of Fire Tables From BLOCKBUSTER

THREAT DIRECT FIRE AND MOVEMENT CAPABILITY (PER TURN)

TANK	PT76	BMP (SAGGER)		BMP (73mm)	MAN-PAC (SAGGER)
S-S-S S-S-M 1/3 S-M 2/3 M 1/3-S-M 1/3 M SM 2 Times	S-S-S S-M 1/2 M 1/2-S M 1/3-S-M 1/3 M	<1000m S-M 1/3 M 1/3-S M S	>1000m S M	S-S-S S-M 1/2 M 1/2-S M M 1/3-S-M 1/3	1 shot per game turn up to 4 times then out of ammo. M

RPG-7	12.7MG	LTMG	ZSU 23-4	SP ARTY	BRDM 14.5mm
S-S S-M 1/2 M M 1/2-S M 1/3-S-M 1/3	S-S S-M 1/2 M 1/2-S M	S-S M S-M 1/2 M 1/2-S	S-S M 1/2-S S-M 1/2 M	S M	S-S M 1/2-S S-M 1/2 M

SNIPER*	PORTABLE FLAME THROWER**
S-S S-M 1/2 M 1/2-S M	S M

Table 5. (Continued)

US DIRECT FIRE AND MOVEMENT CAPABILITY (PER TURN)

TANK	CEV	SP ARTY	TOW <1000M	>1000M	DRAGON
S-S-S S-S-M 1/3 S-M 2/3 M M 1/3-SM 1/3 SM 2 times	S-S M S-M 1/2 M 1/2-S	S M	S-M 1/3 M 1/3-S M S	S M	S M

LAW (PER MAN)	106RR	90RR	LT MG	VULCAN	M203
S S-M 1/2 M 1/2-S M	S S-M 1/2 M 1/2-S M	S-S S-M 1/3 M 1/3-S M	S-S M 1/2-S S-M 1/2 M	S-S M 1/2-S S-M 1/2 M	S-S-S M 1/2-S S-M 1/2 M

PORTABLE FLAME THROWER[1]	SNIPER[2]	.50 CAL (VEH MTD)	FLASH[3]
S M	S-S S-M ½ M ½-S M	S-S S-M 1/2 M 1/2-S SM 2 times	S M

NOTES: SHOOT—S: Applies both to main gun and cupola-mounted machinegun.
MOVE—M: Element may move allotted allowance.
SHOOT on MOVE—SM: Element may shoot on the move one or two rounds (burst).
SHOOT then MOVE—S-M: The weapon system can shoot one round (burst) and then move the allotted movement allowance.

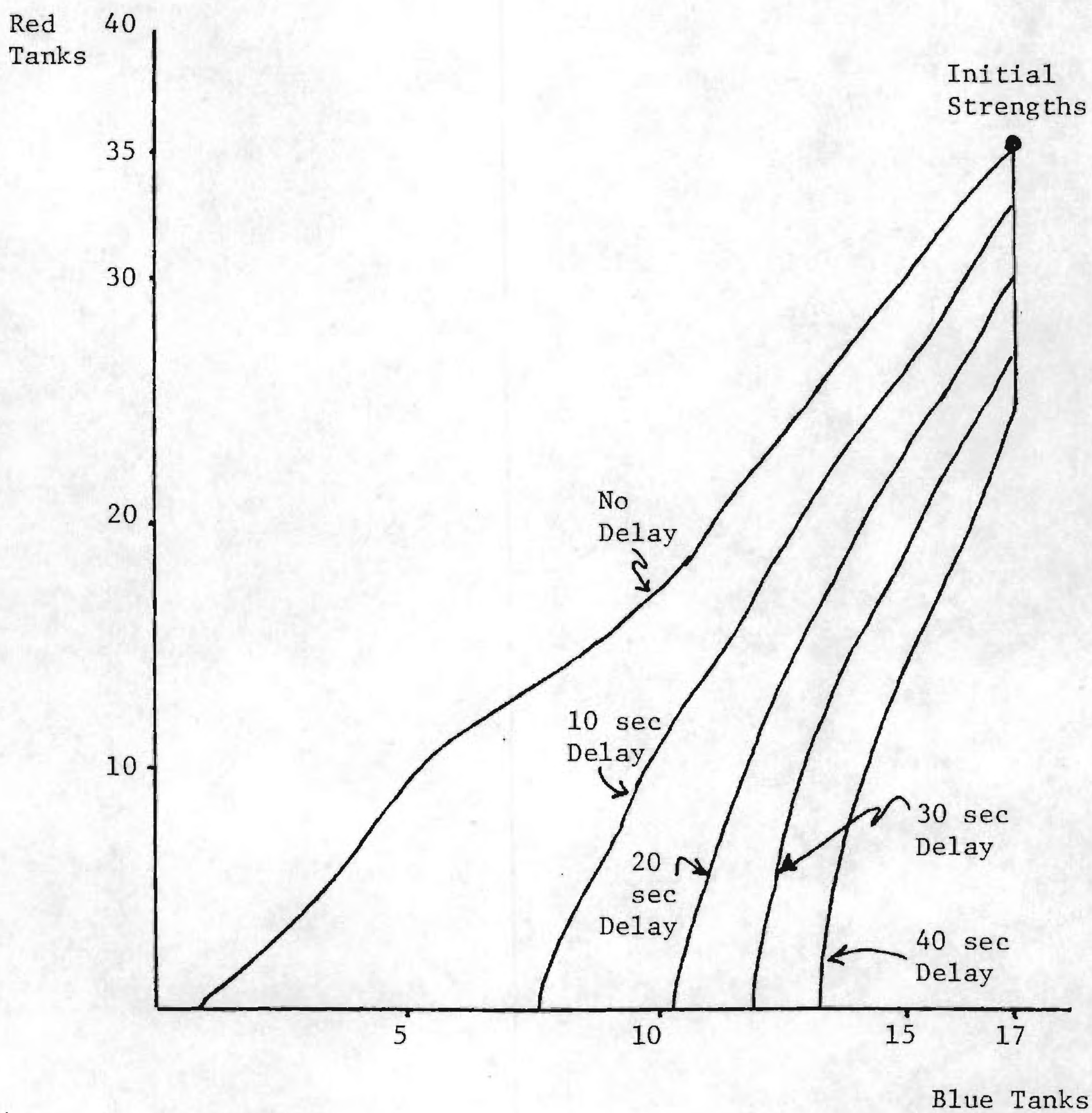


Figure 12. Reaction Time Delays 2200m initial Engagement Range, Red-on-Blue Attrition Delayed; Kill Rates Derived from Bonder's Equation -- From Reference [25]

Whomever fires first can have a significant impact on the outcome of a battle. In a discrete-event simulation, such as ACABUG, it is possible to generate next engagement times based on some known or assumed probability distribution to simulate this occurrence.

By assuming that generating a uniform (0,1) random variable for each engagement opportunity and rank ordering these engagement opportunities based on this random variable it is possible to adequately model this issue in a combined continuous-discrete simulation.

Math Model. Using a combined continuous-discrete simulation, modeling the who fires first question is proposed in the following manner. Direct fire occurs only if an observer acquires a target to which it can trace line of sight and the unit has orders to engage that target. Acquisition attempts are made at the end of each ten-second time step. First, generate a uniform random variable between zero and one for each engagement opportunity. Second, these engagement opportunities are then rank ordered based on their uniform (0,1) random variable and placed in an engagement queue. Third, the engagements in the engagement queue are then resolved sequentially. This sequential resolution could result in observers not conducting their direct fire engagement having been killed in a previous engagement; also in observers firing at a target which has already been killed in a previous engagement. The occurrence of either of these two events in no way detracts from the validity of the war game.

Probability of Detection/Acquisition

The BLOCKBUSTER war game system provides a minimum required distance for the detection/acquisition of a target given the target's disposition and visibility conditions. Probabilities of detecting/acquiring a target beyond this range are assigned based on whether the target is attacking or defending. This probability of detection/acquisition remains constant regardless of the amount of time line of sight is maintained.

Assuming that the length of time required to detect/acquire a target follows an exponential distribution with some known rate, λ, and that target is a candidate for a detection/acquisition attempt provides us with a mathematical model for the BLOCKBUSTER data.

Math Model. Using a probabilistic model discussed by Ross [25]:

$$P(X \varepsilon (t,t+dt) \mid X > t) \approx \frac{f(t)dt}{1-F(t)}$$

$$= r(t)dt$$

where r(t) is the failure (or hazard) rate function and is defined to be:

$$r(t) = \frac{f(t)}{1-F(t)}$$

By assuming the length of time to acquire a target is distributed exponentially, reduces to the constant λ and the result is that $P(X \in (t,t+dt) | X > t)$ reduces to λdt, dependent only on the length of the interval dt and not on the amount of time LOS to the target has been maintained.

CHAPTER IV

EVALUATION AND VALIDATION

Evaluation

Callahan [22] provides three questions which show one method for addressing the issue of model (or in our case, war game) evaluation.

1. Are the models you are using tools or toys?
2. Are you using a man to a boy's job (Is it cost effective?)
3. Does anyone else understand your model?

Although these three questions may be an oversimplification of the issue, the provide some possible general guidelines for the evaluation of a war game.

The issue which must be addressed in this section is that of how to evaluate a computer-driven training war game developed using the methodology described in this thesis. How is this method of training junior leaders in MOUT better than other currently existing methods? Expanding on the questions proposed above we shall investigate this issue and propose a method for conducting such an evaluation.

The evaluation of the usefulness of a war game is not an issue which should be left solely to the model builder. Low [18] provides insight as to why.

> "Another factor is worthy of attention -- the strong competitive spirit within the model building community. At times this spirit stridently manifests itself in the "selling" of a particular model's attributes over those of all rivals. Since there is no way to prove the superiority of one model over another, it is helpful in this environment to point out the shortcomings in competitor's offering."

Evaluation is a matter to be addressed by the decision makers who wish to use this war game based on their intended use.

Catalogs and inventories providing descriptions of the model, simulation, or war game are used by these decision makers to assist in their evaluations of which model would suit their needs. One such catalog is "Catalog of Wargaming and Military Simulation Models", prepared for the Studies, Analysis and Gaming, Organization of the Joint Chiefs of Staff in May 1982. A sample entry on the ARTBASS war game is provided in Table 6.

The most extensive such catalog is the "Inventory of TRADOC Models". This catalog provides a great deal of information on a variety of war games. A sample entry on the ARTBASS war game is provided in Table 7.

One can see, a great deal of information is providided the decision maker to assist in his evaluation of his various choices. A method for evaluating the proposed war game can be based on some of the entries found in this catalog. The author approaches this evaluation from the point of view of a decision maker who is evaluationg ACABUG, ARTBASS, BLOCKBUSTER, and this new war game system, call it MICRO-URB, for use as a means of training junior leaders in MOUT.

Table 6. "Catalog of Wargaming and Military Simulation Models", Sample Entry

TITLE: ARTBASS - Army training Battle Simulator System

PROPONENT: Combined Arms Training Development Activity, Fort Leavenworth, Kansas

DEVELOPER: Combined Arms Training Development Activity

PURPOSE: Through use of a real time battle simulation and a computer graphics display system a battalion commander and staff may be exercised in the command and control realities that will be encountered on the modern integrated battlefield. Permits battalion commander to observe and evaluate ability of his staff to respond to input normally received from subordinate units on a tactical situation. Allows for alternate courses of action to be exercised and evaluated for effectiveness.

GENERAL DESCRIPTION: Lanchester theory used to drive weapons effects, unit attrition, expected values used to determine unit movement, equipment performance curve fit for determining levels of suppression probability theory in line of sight, maintenance factions, etc.

INPUT:

- o Order of battle
- o Firing rates
- o Kill probabilities
- o Mobility
- o Terrain and weather
- o Specific unit order
- o Firing commands

OUTPUT:

- o Sides display of unit locations and battlefield control information
- o Real-time CRT output reports of unit battlefield activity
- o Summary listings over time describing unit status

HARDWARE: Perkin Elmer

SOFTWARE:

- o Programming Language: FTN, some assembler
- o Documentation: Pending

TIME REQUIREMENTS: Pending

SECURITY CLASSIFICATION: UNCLASSIFIED

FREQUENCY OF USE: Under development

USERS: BSD (under development)

POINT OF CONTACT: Herb Westmorland
Combined Arms Training Development Activity
Fort Leavenworth, Kansas
AUTOVON 684-4528

MISCELLANEOUS: Nuclear chemical package and logistic play

KEYWORD LISTING: Computerized, Analytical, Damage assessment, Tactical, Real-time, BN Command and Control Trainer

Table 7. "Inventory of TRADOC Models", Sample Description

TITLE: ARTBASS - Army Training Battle Simulator System

RESPONSIBLE AGENCY: CAORA

PURPOSE: Through use of a real time battle simulation and a computer graphics display system a battalion commander and staff may be exercised in the command and control realties that will be encountered on the modern integrated battlefield. Permits battalion commander to observe and evaluate ability of his staff to respond to input normally received from subordinate units on a tactical situation. Allows for alternate courses of action to be exercised and evaluated for effectiveness.

GENERAL DESCRIPTION: Lanchester theory used to drive weapons effects, unit attrition, expected values used to determine unit movement, equipment performance curve fit for determining levels of suppression probability theory in line of sight, maintenance factions, etc.

INPUT:
- Order of battle
- Firing rates
- Kill probabilities
- Mobility
- Terrain weather
- Specific unit order
- Firing commands

OUTPUT:
- Sides display of unit locations and battlefield control information
- Real-time CRT output reports of unit battlefield activity
- Summary listings over time describing unit status

MODEL LIMITATIONS:
- 100 units
- Can play in any terrain area after the input terrain files are preprocessed

HARDWARE: Perkin Elmer

SOFTWARE: Programming language: FTN, some assembler

TIME WEEKS: Unknown. Under development

Table 7. (Continued)

SECTION I. INFORMATION TO BE FURNISHED ON ATTRIBUTES

Name of Model ARTBASS

Responsible Agency BSD, CACDA

1. Responsiveness.

   a. Time to incorporate a new scenario into the model: *
   b. Time to incorporate major changes in the scenario into the model: *
   c. Time to incorporate minor changes in scenario into the model: *
   d. TIme to incorporate a new system or system change into the model: *
   e. Completeness of documentation:
      Complete: Yes X ; Partial ____ ; No ____
      If answer is "partial" or "no", describe actions being taken to complete the documentation.
   f. Construction, Availability, Maintenance of Data Base:

      (1) Number persons assigned to data base administration of function cell: Number 3 (15)
      (2) Automated data management system used: Yes X ; No ____
      (3) Data dictionary employed: Yes X ; No ____
      (4) Frequency of data base update: Time interval monthly
      (5) How do models interface wtih data: Direct ____ Indirect ____
      (6) Data base related to other data bases: Yes X ; No ____
          If yes, identify. CATTS
      (7) Principal data sources: ____
      (8) Data classification breakdown: TS ____ % C ____ % U 100 %
      (9) Special sensitive data problems: Yes ____ ; No X
      (10) Date transmittal percentage and means: Electronic X % Cards ____ %
           Tape X % Printout X %
           Disk X % Other ____ %
      (11) Time elapsed between requesting date or update and available from base: Average 1 min.
           Shortage ____
           Longest ____

   g. Modular construction: Yes X ; No ____
   h. Computer run times: Run Time 1 day
      Describe parameters on which run time is based. Length of exercise
   i. Total time from initiation of model use to production of results: 3 min
      Describe parameters on which time is based. ____
   j. Post processor with user friendly output. Yes ____ ; No X
   k. Transferable to varied hardware configurations:
      Yes X ; No ____ ; Yes, with difficulty ____

*Free play, INTERACTIVE model, Scenario dependent on player inputs.

Table 7. (Continued)

2. Flexibility.

a. Either a combat developments or a training application with appropriate modification: Training model Yes X ; No ____

b. Different Levels of Intensities of Warfare:

(1) Intense Combat in Main Battle Area: Yes X ; No ____
(2) Light Combat as in screening force or economy of force action: See Note 1 Yes X ; No ___; UD ___; NP ___
(3) Nuclear Warfare: Yes X ; No ___; UD ___; NP ___
(4) Chemical Warfare: Yes X ; No ___; UD ___; NP ___
(5) Rear Area Operations: Yes X ; No ___; UD ___; NP ___

c. Varied Tactics:

(1) Frontal Attack: Yes X ; No ___; UD ___; NP ___
(2) Flanking Attack: Yes X ; No ___; UD ___; NP ___
(3) Meeting Engagement: Yes X ; No ___; UD ___; NP ___
(4) Screening Operation: Yes X ; No ___; UD ___; NP ___
(5) Position Defense: Yes X ; No ___; UD ___; NP ___
(6) Active Defense: Yes X ; No ___; UD ___; NP ___
(7) Spoiling Attack: Yes X ; No ___; UD ___; NP ___
(8) Retrograde Operations: Yes X ; No ___; UD ___; NP ___
(9) Vertical Envelopment: Yes X ; No ___; UD ___; NP ___
(10) Others: Describe:

d. AirLand Battle:

(1) See Deep: Yes X ; No ___; UD ___; NP ___
How Deep 100 Km
(2) Shoot Deep: Yes X ; No ___; UD ___; NP ___
How Deep 100 Km
(3) Discontinuous Line of Contact:
Yes X ; No ___; UD ___; NP ___
(4) Nuclear Warfare: Yes X ; No ___; UD ___; NP ___
(5) Chemical Warfare: Yes X ; No ___; UD ___; NP ___
(6) Integrated Air Ground Operations:
Yes X ; No ___; UD ___; NP ___

e. Difficult Terrain:

(1) Deserts: Yes X ; No ___; UD ___; NP ___
(2) Mountains: Yes X ; No ___; UD ___; NP ___
(3) River Crossings: Yes X ; No ___; UD ___; NP ___
(4) Forests and Jungles: Yes X ; No ___; UD ___; NP ___
(5) MOUT: Yes X ; No ___; UD ___; NP ___
(6) Snow: Yes ___; No X ; UD ___; NP ___
(7) Other: Describe: See note 2

1. To be incorporated into production model. Phase I nuc/chem has been developed for CATTS.
2. Terrain images computer generated based upon DMA supplied digtized terrain data tapes.

Table 7. (Continued)

f. Critical areas of the world:

(1) Europe: Yes X ; No ___; UD ___; NP ___
(2) SE Asia: Yes ___; No X ; UD ___; NP ___
(3) NE Asia: Yes ___; No X* ; UD ___; NP ___
(4) SW Asia: Yes X ; No ___; UD ___; NP ___
(5) Mid East: Yes X ; No ___; UD ___; NP ___
(6) Other: Describe: Ft Irwin/NTC. See note 2 page 2. *Korea data tape to be provided Dec 82.

g. Varied levels of command:

(1) Highest level which can be played: Div
(2) Lowest level which can be played: Bn
(3) Number of echelons which can be played in a single game: 2

3. Consistency. Not yet fielded - currently based on CATTS.

a. Logical Response to Change: Describe sensitivity analyses or other control measures which describe the way the model responds to change.

b. Similar situations produce similar results: Describe sensitivity analyses or other control measures which indicate that model results at worst show only variations when examining similar situations.

c. Audit trail to show effects of change: Describe briefly audit trail mechanisms.

4. Rapid data access.

a. Direct user inputs: Yes X ; No ___; UD ___; NP ___
b. Rapid data access by electronic means or tapes:
Yes X ; No ___; UD ___; NP ___
c. Direct inputs from one model to another:
(1) To higher echelon models: Yes ___; No X ; UD ___; NP ___
(2) To lower echelon models: Yes ___; No X ; UD ___; NP ___
(3) From higher echelon models: Yes ___; No X ; UD ___; NP ___
(4) From lower echelon models: Yes ___; No X ; UD ___; NP ___

d. Automation of the Data Base: Yes X ; No ___; UD ___; NP ___

e. Standardization of data base structures:
(1) With other models: Yes X ; No ___; UD ___; NP ___
(2) With data suppliers: Yes X ; No ___; UD ___; NP ___

5. Credibility.

a. Time length and coverage: Not yet fielded

(1) Average length of game time: 8 hrs.+
(2) Size of area covered: 3000 Km

+Game-time based on using unit training needs.

Table 7. (Continued)

b. Play of Arms and Services: Game time based on using unit training needs.

(1) All Combat Arms: Yes X ; No ___; Note Exceptions.
(2) All Combat Support Arms: Yes X ; No ___; Note Exceptions.
(3) Total Logistics: Yes X ; Part ___; No ___; UD ___; NP ___
(4) Air Force: Yes X ; No ___; UD ___; NP ___
(5) Civilians: Yes ___; No X ; UD ___; NP ___

c. Play of Allied Forces:

(1) Europe: Yes ___; No X ; UD ___; NP ___
(2) Mid East: Yes ___; No X ; UD ___; NP ___
(3) Other: Decribe: No NATO or other forces modeled.

d. Play of Varied Dispositions:

(1) Continuous Defenses: Yes X ; No ___; UD ___; NP ___
(2) Discontinuous Defense: Yes X ; No ___; UD ___; NP ___
(3) Very wide coverage delay and defense:
Yes X ; No ___; UD ___; NP ___
(4) Multiple Attacks: Yes X ; No ___; UD ___; NP ___

e. Play of Contingency Operations:

(1) Deployability considered: Yes ___; No X ; UD ___; NP ___
(2) Light Forces Examined: Yes X ; No ___; UD ___; NP ___
(3) Principal contingency areas played: Yes ___; No ___
If no, state what can be played. (See note 2 page 2)

f. Play of Degraded Environments:

(1) Smoke: Yes X ; No ___; UD ___; NP ___
(2) Dust, Haze: Yes X ; No ___; UD ___; NP ___
(3) Fog, Sleet, Rain: Yes X ; No ___; UD ___; NP ___
(4) Jamming: Yes ___; No X ; UD ___; NP ___
(5) Chaff: Yes ___; No X ; UD ___; NP ___

g. Consistent with Test and Experiments:

Discuss the way test data influences the design and use of the model. All new module designs are based on Army publications or coordination with Army agencies responsible for functional area.

h. Play of realistic threat systems and tactics.

(1) Red tactics and systems played in accordance with threat information. Yes X ; No ___; UD ___; NP ___
(2) Asymmetric play (blue tactics by blue, red tactics by red).
Yes X ; No ___; UD ___; NP ___

i. Complete two-sided battle. Yes X ; No ___
If no, decribe limitations.

Table 7. (Continued)

j. Smallest unit to which resolved.

Item ____; Squad _X_; Platoon ____; Company ____; Brigade ____; Division ____; Corps ____; Theater ____.

6. Transparency. Describe the steps taken to inform the decisionmaker, trainer and/or controller of the impact that model routines have on model outputs.

7. Resources.

a. Number of full time personnel required to set up, operate and maintain model: __8__
b. Number of full time personnel required to set up and maintain data base: __15__
c. State computers on which it can operate, and describe demands upon those computers in terms of core requirements, input/output devices and other matters of significance. Perkin Elmer
d. Discuss model efficiency in. terms of the interaction of routines.
e. Degree of automation.

Fully automated:	Yes X	No ____	UD ____	NP ____
Man in the loop:	Yes X	No ____	UD ____	NP ____
Use of order streams:	Yes ____	No X	UD ____	NP ____
"Top Down" decisionmaking (as in DYNTACS):	Yes ____	No X	UD ____	NP ____
Input Decision tables:	Yes ____	No X	UD ____	NP ____

f. Capability for self documentation of model routines:
Yes X; No ____; UD ____; NP ____

8. Functional Coverage. Indicate ability to play functional area indicated.

a. Close combat (L) including dismounted operation:	Yes X	No ____	UD ____	NP ____
b. Close combat (H):	Yes X	No ____	UD ____	NP ____
c. Fire Support:	Yes X	No ____	UD ____	NP ____
d. Engineer and mine warfare:	Yes X	No ____	UD ____	NP ____
e. NBC:	Yes X	No ____	UD ____	NP ____
f. Air defense:	Yes X	No ____	UD ____	NP ____
g. Intel EW:	Yes ____	No X	UD ____	NP ____
h. CSS:	Yes X	No ____	UD ____	NP ____
i. Aviation:	Yes X	No ____	UD ____	NP ____
j. Battlefield nuclear:	Yes X	No ____	UD ____	NP ____
k. Communications:	Yes ____	No X	UD ____	NP ____
l. Command and control:	Yes X	No ____	UD ____	NP ____

m. Is coverage of functions balanced? Discuss. Yes
n. Can detailed examinations of all or some of the functional areas be made within the context of the overall model (Variable resolution)? Discuss. *See note page 1

Table 7. (Continued)

SECTION II. ADDITIONAL DATA

9. Model limitations. Discuss model limitations not covered in the preceeding paragraphs. 200 units.

10. Treatment of change events. Deterministic _X_ Stochastic _X_

11. Treatment of simulated time. Time Step _X_ Event Sequence ___

12. Describe how terrain is modeled. See note 2 page 2

13. Describe primary means for assessing attrition.
 X Differential (Lanchester) acquisitions
 _ One on one examinations
 _ Statistical basis
 X Input parameters
 _ Other (Describe)

14. List principal data requirements. For example.
 _ Organizations
 _ Tactics
 X Kill probabilities
 _ Mobility
 X Terrain and Weather

15. List principle outputs. For example.
 X Computer printout with Post processor
 _ Computer printout with raw data
 _ Plots
 _ Tables
 _ Statistics
 _ FEBA Movement
 _ Time periods

16. Hardware.
 _ Computer (Manufacturer and Model). Perkin Elmer 5240
 _ Operating System. EXEC P
 _ Minimum Storage Required. 2.5 megawords
 _ Peripheral Equipment. printer, disc, terminals, graphic display devices
 _ Color Graphics. Lexadata

17. Software.
 _ Programing Languages. FORTRAN
 _ Documentation. List by title, publication, date and DTIC accession number if applicable. For documents not published, indicate if draft copies are available and expected date of publication. IAW MIL-STD-1644

18. Security Classification of model less data. Unclassified

19. Frequency of use. 250 days a year

Table 7. (Continued)

20. Organizations which have used output. ARI, TRASANA

21. Point of contact.

__ Name. MAJ DELLOYD VOORHEES JR.
__ Organization with office symbols. CACDA, BSD, ATZL-CTB-T
__ Address. Fort Leavenworth, KS 66027
__ Telephone number. AV 552-3180/3395

Upon examining the attributes described in the "Inventory of TRADOC Models" eight attributes were selected which are key to the evaluation of a war game which is to be used to train junior leaders in MOUT. These attributes are defined below and used in a decision matrix shown in Table 8.

1. "Either a Combat Development or a Training Application with Appropriate Modification" -- Can this war game be used for training?
2. "Difficult Terrain: MOUT" -- Is MOUT portrayed at the level required?
3. "Varied Levels of Command: Lowest Level Which Can Be Played" -- Can this model be used to train my junior leaders?
4. "Average Length of Game Time" -- How long to play one game? (This is very much scenario dependent. An attempt should be made to compare a ratio of real time to game time if this information is available).
5. "Smallest Unit to Which Resolved" -- Does the war game allow the players to address units which they would commonly address in combat?
6. "Number of Full-Time Personnel Required to Set Up, Operate, and Maintain Model" -- How many people will be required to play this game?
7. "Model Limitations" -- What limitations to this war game does the model builder recognize?

Table 8. A Decision Matrix for ACABUG, ARTBASS, BLOCKBUSTER and MICRO-URB

ATTRIBUTE / WAR GAME	(1) TRAINING APPLICATION	(2) MOUT REPRESENTED	(3) LOWEST ECHELON PLAYED	(4) AVERAGE LENGTH OF GAME (REAL:PLAY)	(5) SMALLEST UNIT TO WHICH RESOLVED	(6) PERSONNEL REQUIRED	(7) MODEL LIMITATIONS	(8) COMMENTS
ACABUG	YES	YES (AS INDIVIDUAL BUILDINGS WITH ROOMS)	INDIVIDUAL SOLDIER	UNKNOWN (DESIRED 1:10)	ITEM	12 - 15 TO OPERATE	WAR GAME UNDER DEVELOPMENT	PRIMARILY ANALYTICAL TOOL; LARGE GAME BOARDS USED
ARTBASS	YES	YES (AS A TER-RAIN TYPE	BATTALION	8+ HOURS (ALMOST 1:1)	SQUAD	8	MAX OF 200 units	VAN MOUNTED FOR MOBILITY
BLOCKBUSTER	YES	YES (AS INDI-VIDUAL BUILDINGS)	SQUAD	8 HOURS (ESTIMATED 1:30)	ITEM	6 (MINIMUM)	NONE LISTED	MANUAL GAME; DUAL SCALE TERRAIN BOARD
MICRO-URB	YES	YES (AS INDI-VIDUAL BUILDINGS)	SQUAD	UNKNOWN	ITEM	2 (ESTIMATED)	WAR GAME UNDER DEVELOPMENT	MICRO-COMPUTER BASED

8. "Comment" -- Any insights which may be useful in making a decision?

This is by no means an exhaustive list and other decision makers may construct their decision matrix in a different manner. Attributes in the decision matrix may be added or deleted based on the decision maker's needs.

Connolly [12] describes the "expectancy model" for making decisions. Using this model the net value of an action or alternative is the sum of the expected value of the outcomes of this alternative weighted by the probability of achieving the outcome. That is

$$\text{Net value of action } i = \sum_{\text{all } j} P_{ij} V_j$$

where

i = number of alternatives

j = number of outcomes

V_j = expected value of the jth outcome

P_{ij} = probability of achieving outcome j by choosing alterative i

The degree of uncertainty about some of the characteristics of MICRO-URB does not allow us to use the expectancy model at this time.

By evaluating the MICRO-URB war game system based on the attributes listed in the decision matrix we can make the following observations.

1. ARTBASS does not allow the players to use urban terrain at the resolution needed to train junior leaders.

2. ACABUG, BLOCKBUSTER and MICRO-URB are similar in most attributes.

3. MICRO-URB provides considerable savings in personnel requirements and may also achieve a significant reduction in required playing time.

4. MICRO-URB is based on a micro-computer and may not require additional facilities to conduct training if the decision maker has the necessary hardware base (i.e., MICROFIX).

MICRO-URB would seem to meet the requirements for training junior leaders in MOUT at a reduced cost when compared to the current methods.

Validation

The issue of the validity of this, or any, war game is a difficult and important one. The assumption which must at some point be validated, if the war game is to be considered a valid one, is the assumption that this war game provides an accurate representation of urban combat for the purposes of training junior leaders in MOUT.

Dunnigan [12] describes his system for validating war games based on historical data. In describing his method for assigning combat values to the units involved in the historical battle he states: "The system is further refined when you start playing the game and any misjudgements you have made quickly become evident in your attempts to recreate the historical event". In modeling a historical battle or campaign, it is necessary that the results of the historical battle be in the realm of possible outcomes produced by the model. It is not necessary that the historical event be at or near the mean outcome of your model, only that it be a possible outcome; afterfall, the historical event is just one realization of the possible results of the battle being modeled.

How then is this or any other model of theoretical combat to be validated. Hoeber [16] discusses at length the structure and purpose of a variety of models developed for, and by, the military. In discussing the validity of these models, Hoeber seems to place a great deal of emphasis on the "face" validity of the model; i.e., "the acceptability of the model by a high-level decision maker"; and an examination of what the model has done under a particular set of circumstances by experienced personnel for confirmation of the reasonableness of the results. He notes that the "face" validity of the model can be modified in two ways: (1) Change the model to obtain performance more acceptable to the decision maker or experts; or (2) Convince the decision maker or experts that the model is a correct representation of reality and that their perception of reality is incorrect.

Neither the "face" validation suggested by Hoeber, nor Dunnigan's implication that a model is valid if the historical results can be reproduced by the model, are totally analytical approaches. Both methods, however, do provide a method for validating a combat model in the absence of complete analytical data.

The tables and charts used by BLOCKBUSTER and other war games have been derived analytically and verified by the U.S. Army Material

Systems Analysis Activity (AMSAA) and the U.S. Army TRADOC Systems Analysis Activity (TRASANA). These represent factors which can be isolated and measured, for example, the effects of a given projectile fired from a given range at a given target. Other factors of the game are not quite so easy to study analytically because they involve humans placed in uncommon situations. For example, what are the effects of suppression and what concentration of fire power is required to suppress a combat unit. Overriding the validity of the individual components of the model is the validity of combining the individual components of the model and having the resulting model adequately describe urban combat.

The individual components and factors present in actual combat combine in a synergistic manner; i.e., the result is not merely the sum of the parts. The very nature of combat does not lend itself to analytical study. Most historical battles do not provide enough information to conduct detailed analysis. Units present, their equipment, their state of readiness, and training, even casuality reports may not exist, or if they do exist, they are not detailed enought to validate a combat model.

Ogus [22] states that the validity of a training war game lies not in how well it simulates reality, but rather in whether or not it teaches the concepts and facts the game designer intended it to teach. This approach to validation is also subjective.

Connolly [12] discusses a concept put forth by Herbert Simon called "satisficing". As used in Connolly's book, this term attempts to describe the organizational decision making process; however, in light of the face validation method described by Hoeber, I feel this term applies to attempts to validate military combat models. Connolly states by way of a definition of satisficing:

> "Herbert Simmon ... suggests that we should think in terms of 'bounded' rather than strict rationality. These 'bounds' are imposed both by our own weaknesses as decision makers and by the difficulties of the situations we have to cope with. We get by reasonably well in most cases by simplifying the decision process. That is: (1) We consider only a few major value dimensions; (2) We examine a few possible alternatives; (3) We consider only some of the main consequences of each alternative; (4) We choose the first alternative that reaches some level we think of as 'good enough'; To emphasize the distinction between this model and the strictly rational process, Simon uses the term 'satisficing' for the boundedly rational process, in contrast to 'optimizing' for strictly rational process."

The goal of any model or simulation should not be that of "satisficing" the decision maker, nor of achieving face validation, rather, the goal should always be to optimize the degree to which the model or simulation represents reality. Law and Kelton [17] address the difficulty of providing a valid simulation in the following statement: "However, the most 'valid' model will not necessarily be the most cost effective. One should always keep in mind the overall objective of the simulation study ...".

The study of war is an inexact science. Hausrath [15] outlines seven points which limit the value of war gaming as an accurate portrayal of combat.

1. An inability to predict how a man will react to combat.
2. The vast number of variables and interactions which exist in combat.

3. The variables do not necessarily recur in fixed amounts of weight of relative importance.
4. Our current understanding of warfare is incomplete.
5. "Break" point cannot be accurately predicted and may vary greatly from battle to battle.
6. The influence of stress, courage, fear, morale, and leadership on combat remains intangible.
7. Even the affects of measurable physical factors, such as the rate and accuracy of fire, on units in combat are largely unknown.

The use of an umpire to input intangible factors into the war game is useful, but the question remains, what guidelines does the referee use.

Does adding additional factors into a war game increase the validity of the war game? It would seem so, but not without a cost. Low [18] has some interesting comments on the danger of adding more detail to the model.

> "Another natural reaction among the model development community also stems from the unsettled, unproven foundations on which computerized war gaming has been built. This is the propensity for the introduction of more and more detail into combat models, usually at the suggestion or insistence of experienced military officers who might argue that without the treatment of some specific factor or other in the model, one does not have a proper representation of the combat process. Just how much of such material might be fundamental in shaping the outcome in a combat model and how much of it might merely go into immortalizing in computer code the reflections of an individual's combat experience is, of course, not known. The analyst may well argue that sensitivity analysis could provide the answer, but more often than not such analyses are

> not conducted. If the suggested additions seem reasonable, or if, perchance, contracts might hang in the balance, then into the model they go.
>
> Of significance here is the almost subliminal pursuit of a goal which, in the absence of any other hand-hold on the problem, reflects the need to converge on reality through the eventual simulation of every finger on a trigger and every round of ammunition fired. As a result, models tend to become more and more complex and their appetite for input data increases commensurately. This, of course, is in direct contradiction to the tenets of systems analysis in which the objective is ultimately to isolate and identify the parameters 'that really matter'; those that have a first order effect on system behavior. Without a laboratory in which to conduct experiments and without any fundamental theoretical knowledge, the traditional systems analysis approach then appears to be unattainable, and we are left somewhere in mid-flight between two worlds, so to speak, without enough computational power, reliable data, or real knowledge about the dimenions of the problem to see us through to our destination. In the meantime, the opacity, complexity, and costs of the models that are being developed are great enough to cause frustration among those who support and use them."

Nonetheless, in the absence of "a laboratory in which to conduct experiments" we must keep in mind the goal of this training war game -- to train junior leaders in MOUT. It is hoped that the additional factors outlined in this thesis add to the overall validity of the model. However, as a training war game an increase in the face validity of the model is also important and we achieve this in the proposed methodology.

CHAPTER V

RESULTS AND RECOMMENDATIONS

Results

In investigating the potential use of micro and mini-computer driven war games for training junior leaders, the entire family of war game models has been research for potential use in improving MOUT training. It was found that data and techniques used by several models could be directly applied to the development of a new computer driven war game, MICRO-URB. In the proposed methodology the BLOCKBUSTER manual war game is used as a basis for determining general attributes which would be desirable in a computer driven urban war game, as well as a data base. ACABUG provides algorithms for suppression determination and building clearance which are directly applicable to MICRO-URB. Additionally, ACABUG provides a tremendous source of analytical data which can be selectively added to the model as memory permits. ARTBASS provides an existing digitized terrain data base and techniques for managing this data which are invaluable to the methodology proposed here. Additionally, ARTBASS has algorithms for terrain appreciation display and the definition of engagement priorities which can be modified for use on the proposed war game. The MICROFIX micro-computer system was investigated as a potential hardware base for the proposed war game. MICROFIX has the capability to display military map sheets overlayed with graphics displays. Additionally, the data base management systems used by MICROFIX are useful in the management of the unit and urban data files which the proposed war game require.

Although research into the family of war games revealed many techniques which could be used directly in the development of this methodology, certain areas were found to be inadequately addressed by these models. Original research into these areas produced the following results:

1. A method for representing urban terrain as an overlay rather than generating a new digitized terrain data base has been proposed. This method makes use of algorithms developed for use in ARTBASS and CATTS to assist in the building of the urban overlay data base and retrieving information from this data base. Additionally, an algorithm was developed for determining the points of intersection of an observer target line and a building so that urban line of sight can be checked.

2. A mathematical model of a common tactic which assigns weapon systems and units sectors of responsibility in which to concentrate their primary detection/acquisition assets has been developed. Use of this model and the algorithm proposed adds to the face validity of the war game by applying this tactic to reduce the amount of battlefield intelligence available to the player. Additionally, by conducting line of sight checks only for targets which fall within a units sector of responsibility we will reduce the time needed to calculate line of sight.

3. The use of combined continuous-discrete simulation techniques is proposed along with suggestions for addressing the probability of detecting/acquiring a target given that line of sight exists. A method for resolving the issue of who fires first is also proposed.

Finally, a method has been developed for evaluating the usefulness of the proposed war game. Using a decision matrix, it is possible to identify the ability of the four war gaming systems being evaluated to provide the attributes which are required to train junior leaders in MOUT. Such a decision matrix can then be used by the decision maker to assist in his evaluation of available assets.

The validity of combat models was also addressed. Techniques currently available provide a subjective means of validating combat models but analytical data is not available to test the validity of these models.

Limitations

The methodology proposed by this research is necessarily limited by the general assumptions outlined in Chapter III. These assumptions, which primarily relate to the manner in which urban terrain and target acquisition are modeled, are necessary to develop the mathematics and dynamics of this methodology. Although the methodology was developed to expand the usefulness of the war game as a tool for training junior leaders in MOUT, it is not intended to be an analytical tool..

It is assumed that the methodology proposed herein does not significantly enhance nor degrade the capabilities of the opposing forces who will be portrayed in this wargame. The effects of such model attributes as sectors of responsibility and the aggregation of buildings into building types on this assumption has not been investigated.

Recommendations

A method for representing urban terrain as an overlay has been proposed. The representation of buildings as rectangular parallelepipeds should be expanded so that buildings may be represented as a more complex union of rectangular parallelpipeds. Additionally, the effect of this more complex modeling on the algorithms proposed herein should be examined.

A method has been proposed which uses a probability of detecting/acquiring a target to determine engagement opportunities. If we assume that the time between target detections/acquisitions is distributed based on a probability distribution other than exponential we may be able to make this probability dependent also on the time since line of sight was first obtained. Additionally, more investigation is needed to determine the affects of a temporary loss of line of sight.

Many commercial war games assign victory points or set victory conditions to allow the player to evaluate his performance. This system demonstrates to the player that it is possible to "win the battle but lose the war". Research into this area could provide a quantitative evaluation of performance to be applied along with the usual subjective evaluation conducted by the trainer.

In the area of sensitivity analysis and model validation a great deal of research remains to be done for models in general and for this specific model. One method of conducting this analysis would be to compare the results of independent runs of the scenario described in Appendix A on the ACABUG, BLOCKBUSTER, and MICRO-URB war game systems.

APPENDIX A

APPENDIX A

This appendix describes a typical scenario which could be played using the computer driven training war game described in this thesis. This scenario is tended to give readers who are unfamiliar with the military an idea of the number of units which would be involved in MOUT operations, the size of the area of operations, and the military objectives of such operations. This scenario was created based on personal experiences and is not intended to establish maximum or minimum parameters of the war game.

A map of a portion of the area of operations for the battle is shown in Figure 1. The blue player is to be a U.S. Army platoon leader. The red player will command a Soviet company.

Forces

AMERICAN MECHANIZED INFANTRY PLATOON (REINFORCED)

- 4 squads -- each squad can be further broken down into 2 fire teams of 4 and 5 men, respectively
- 4 armored personnel carriers (APC's) -- each APC can carry one squad and mounts a .50 caliber machine gun
- 4 M-60 medium machine guns -- must be assigned to a specific fire team
- 2 DRAGON - MEDIUM RANGE, ANTI-TANK, GUIDED MISSILES -- must be assigned to a specific fire team
- 2 TOW's-LONG RANGE, ANTI-TANK WIRE GUIDED MISSILE -- mounted in APC's these come with their own crews

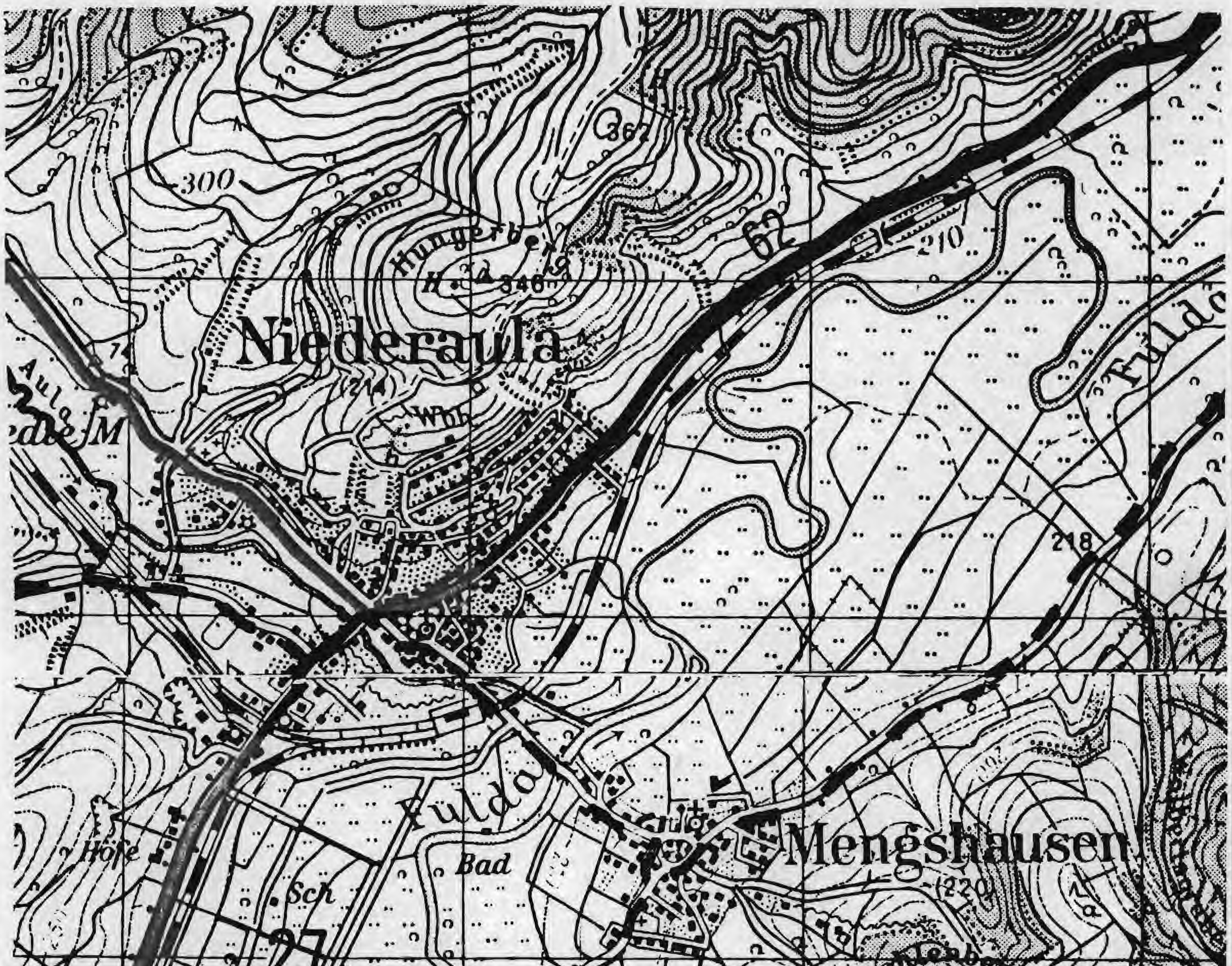


Figure 1. Scenario Area of Operations

SOVIET MOTORIZED RIFLE COMPANY (REINFORCED)

- 9 squads -- 9 to 12 men, Soviet tactical doctrine does not usually allow the squad to be broken down into fire teams
- 9 armored personnel carriers -- each can carry one squad and mounts a SAGGER, long range, anti-tank wire guided missle; a 73 mm cannon and a medium machine gun
- 9 medium machine guns -- must be assigned to a specific squad
- 3 tanks -- each equipped with a main gun and two machine guns

Mission

American

Platoon is to deploy in the vicinity of NIEDERAULA to defend against a possible attack by a Soviet force attempting to secure a bridgehead across the Fulda River. Attack is most likely to be by a reinforced motorized rifle company entering your area of operations from the east or southeast. Platoon should be prepared to withdraw along the Aula River Valley on order. Platoon is to avoid becoming decisively engaged at all costs.

Soviet

Attack from the south and southeast to secure a bridgehead across the Fulda River in the Niederaula-Mengshausen vicinity. Secure crossing site and high ground west of the Fulda River to insure safe crossing of follow-on units. American resistance is expected to be light.

Training Objectives

Train the platoon leader to task organize based on a company operations order. The ability of the blue player to use urban terrain allows him to incorporate the village of Niederaula into his defensive plan. His orders, not to become decisively engaged, are likely to preclude him from conducting a deliberate defense of this village.

APPENDIX B

COMPUTER CODE FOR ELEV SUBROUTINE

Variable List

DIMEN%	- standard terrain square dimension
BL(i,j); j=1,2	- (x,y) coordinates of building corners (in world coordinates)
EL(i,j)	- known elevation of point (i,j) i,j in normalized coordinates
XB%, YB%	- normalized (x,y) coordinates of terrain square containing corner point
XU%, YU%	- normalized (x,y) coordinates of square above and left of terrain square containing corner point
Z1, Z2, Z3, Z4	- elevation at the four corners of the terrain square containing the building corner point
XC%, XCT%, YC%, YCT%	- XB%, XU%, YB%, YU% converted to world coordinates
A1, B2, C3, D4	- see Math Model in Chapter III
ELV(i)	- elevation of ith corner point

```
6095 '************************************************************************
6096 '***                          ELEV SUBROUTINE                          ***
6097 '************************************************************************
6100 DIMEN%=25'  TERRAIN SQUARE DIMENSION
6110 SQD%=DIMEN%^2
6120 DIM EL(27,9)
6130 DATA 640,185,655,195,645,210,630,200
6140 DATA 214,216,214,218,219,217,219,220,221
6150 FOR I=1 TO 4
6160 FOR J=1 TO 2
6170 READ BL(I,J)'   READS (X,Y) COORDINATES OF BUILDING CORNERS
6180 NEXT J
6190 NEXT I
6200 FOR I=25 TO 27
6210 FOR J=7 TO 9
6220 READ EL(I,J)'   READS KNOWN ELEVATIONS
6230 NEXT J
6240 NEXT I
6250 '************************************************************************
6251 '***                          MAIN PROGRAM                          ***
6252 '************************************************************************
6255 S$=TIME$
6260 FOR I=1 TO 4'  FOR EACH OF THE BUILDING CORNERS
6270 XB%=FIX(BL(I,1)/DIMEN%):YB%=FIX(BL(I,2)/DIMEN%)'  DETERMINES THE NORMALIZED
 (X,Y) COORDINATES OF THE SQUARE
6280 XU%=XB%+1:YU%=YB%+1' SETS UPPER BOUNDARIES OF SQUARE
6290 Z1=EL(XB%,YB%):Z2=EL(XU%,YB%):Z3=EL(XB%,YU%):Z4=EL(XU%,YU%)'  ASSIGNS KNOWN
 ELEVATIONS TO THESE VARIABLES
6300 XC%=XB%*DIMEN%:XCT%=XU%*DIMEN%:YC%=YB%*DIMEN%:YCT%=YU%*DIMEN%'  CONVERTS NO
RMALIZED COORDINATES TO MAP COORDINATES
6310 A1=Z1-Z2-Z3+Z4'   SEE MATH MODEL
6320 B2=(YCT%*(Z2-Z1))+(YC%*(Z3-Z4))'   SEE MATH MODEL
6330 C3=(XCT%*(Z3-Z1))+(XC%*(Z2-Z4))'   SEE MATH MODEL
6340 D4=(XCT%*((YCT%*Z1)-(YC%*Z3)))+(XC%*((YC%*Z4)-(YCT%*Z2)))'  SEE MATH MODEL
6350 ELV(I)=((A1*BL(I,1)*BL(I,2))+(B2*BL(I,1))+(C3*BL(I,2))+D4)/SQD%'  DETERMINE
S ELEVATION OF POINT I
6360 NEXT I
6365 F$=TIME$
6370 FOR I=1 TO 4
6380 LPRINT "CORNER POINT # "I" ELEVATION IS "ELV(I)
6390 NEXT I
6395 LPRINT S$,F$
6400 END
```

```
CORNER POINT #  1  ELEVATION IS  216.96
CORNER POINT #  2  ELEVATION IS  219
CORNER POINT #  3  ELEVATION IS  217.6
CORNER POINT #  4  ELEVATION IS  216.6
01:49:25       01:49:25
```

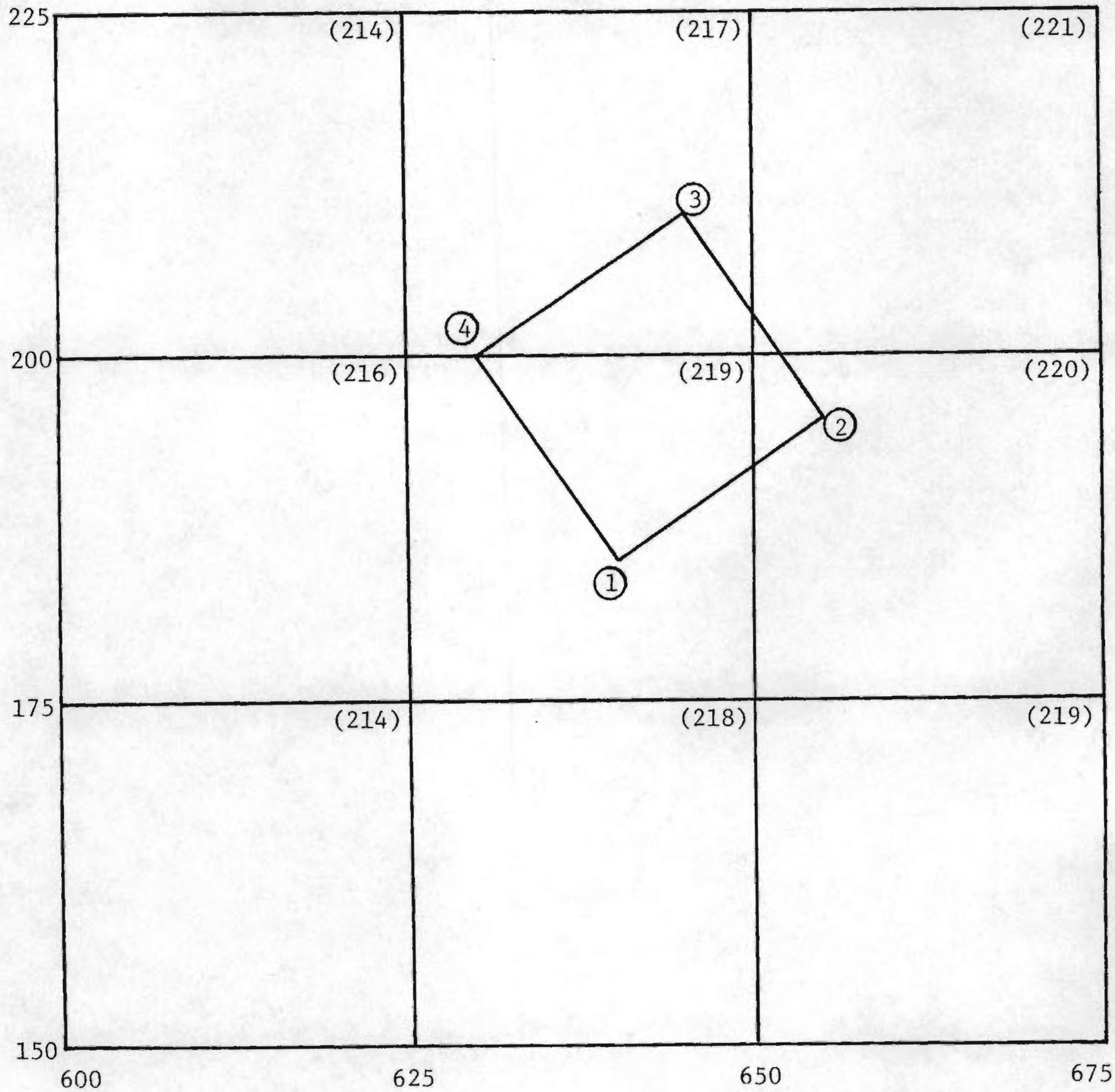


Figure 2. Sketch of ELEV Data

APPENDIX C

COMPUTER CODE FOR BOXES SUBROUTINE

Variable List

X1,Y1	- (x,y) observer coordinates
X2,Y2	- (x,y) target coordinates
DIMEN%	- dimension of box
NZ%	- number of boxes in a row
K%	- number of shotlines
XSTART%, YSTART%	- starting grid block coordinates (normalized)
XSLOPE%	- slope of normalized line in X direction
YSLOPE%	- slope of normalized line in Y direction
SIGNX%, SIGNY%	- sign of slope in X and Y directions, respectively
XCUR%, YCUR%	- current normalized X and Y grid line being crossed by the OT line, respectively
AB%(i,j)	- stores normalized X and Y coordinates of the grid block crossed
DXY, DX, DY%, Q%, R%	- see Math Model in Chapter III

```
4997 '************************************************************************
*
4998 '***                      BOXES  SUBROUTINE                          ***

4999 '************************************************************************

5000 DIM AB%(20,3)'  DIMENSIONS ARRAY TO STORE ZONES CROSSED
5010 DIMEN%=25:K%=10:NZ%=4:N%=1'  SETS SIZE OF BOX=25,NUMBER OF SHOTLINES=10,AND
 NUMBER OF BOXES IN A ROW=4
5015 S$=TIME$
5020 FOR I=1 TO K% STEP 2
5030 READ X1,Y1,X2,Y2'  READS IN OBSERVER AND TARGET COORDINATES
5040 XSTART%=FIX(X1/DIMEN%):YSTART%=FIX(Y1/DIMEN%):XFINISH%=FIX(X2/DIMEN%):YFINI
SH%=FIX(Y2/DIMEN%)'  DETERMINES THE OBSERVER AND TARGET BOX COORDINATES
5050 XSLOPE%=XFINISH%-XSTART%:YSLOPE%=YFINISH%-YSTART%'  DETERMINES THE NUMBER O
F BOXES CROSSED IN THE X AND Y DIRECTIONS RESPECTIVELY
5090 SIGNX%=SGN(XSLOPE%)'  DETERMINES THE SIGN OF THE SLOPE IN X DIRECTION
5100 SIGNY%=SGN(YSLOPE%)'  DETERMINES THE SIGN OF THE SLOPE IN Y DIRECTION
5110 IF SIGNX%=-1 THEN Q%=0 ELSE Q%=1'  SEE MATH MODEL
5120 IF SIGNY%=-1 THEN R%=0 ELSE R%=1'  SEE MATH MODEL
5130 N%=N%+1:AB%(N%,1)=XSTART%:AB%(N%,2)=YSTART%:AB%(N%,3)=I'  PLACES STARTING B
LOCK IN ZONE CROSSED ARRAY
5140 XCUR%=XSTART%*DIMEN%:YCUR%=YSTART%*DIMEN%:DXY=((X2*Y1)-(Y2*X1)):DX=X2-X1:DY
=Y2-Y1'  INITIALIZES VALUES OF XCUR AND YCUR IN GLOBAL COORDINATES AND COMPUTES
THE VALUES OF DXY,DX,AND DY
5160 IF XSLOPE%=0 GOTO 5220' NO BOXES CROSSED IN THE X-DIRECTION CHECK Y DIRECTI
ON
5170 FOR J=1 TO ABS(XSLOPE%)'  COMPUTE INTERSECTION FOR EACH BOX CROSSED IN THE
X DIRECTION
5180 N%=N%+1:XCUR%=(XCUR%+(DIMEN%*SIGNX%*Q%)):Q%=1' COMPUTE VALUE OF XCUR
5190 IF SIGNX%=-1 THEN AB%(N%,1)=(XCUR%/DIMEN%)-1 ELSE AB%(N%,1)=XCUR%/DIMEN%'
DETERMINES THE X COORDINATE OF THE BOX BEING CROSSED AND STORE IN THE BOXES CROS
SED ARRAY
5200 AB%(N%,2)=FIX(((XCUR%*DY+DXY)/DX)/DIMEN%):AB%(N%,3)=I'  COMPUTE Y INTERSECT
ION AND STORES IN THE BOXES CROSSED ARRAY
5210 NEXT J
5220 IF YSLOPE%=0 GOTO 5280'  NO BOXES CROSSED IN Y DIRECTION, PRINT BOXES CROSS
ED ARRAY
5230 FOR J=1 TO ABS(YSLOPE%)'  COMPUTE INTERSECTION FOR EACH BOX CROSSED IN THE
Y DIRECTION
5240 N%=N%+1:YCUR%=(YCUR%+(DIMEN%*SIGNY%*R%)):R%=1' COMPUTE VALUE OF YCUR
5250 IF SIGNY%=-1 THEN AB%(N%,2)=(YCUR%/DIMEN%)-1 ELSE AB%(N%,2)=YCUR%/DIMEN%'
DETERMINES THE Y COORDINATE OF THE BOX BEING CROSSED AND STORES IN THE BOXES CRO
SSED ARRAY
5260 AB%(N%,1)=FIX(((YCUR%*DX-DXY)/DY)/DIMEN%):AB%(N%,3)=I'  COMPUTE X INTERSECT
ION AND STORE IN THE BOXES CROSSED ARRAY
5270 NEXT J
5280 NEXT I
5283 F$=TIME$
5285 FOR T=2 TO N%'  PRINT OUT THE RESULTS
5290 LPRINT "LINE"AB%(T,3)"INTERSECTS ZONE "AB%(T,1)+1+(NZ%*AB%(T,2))'  COMPUTES
 BOX NUMBER CROSSED
5300 NEXT T
5310 LPRINT S$,F$
5320 DATA 15,65,65,20,80,20,60,70,40,90,45,95,85,70,95,30,5,15,30,40
5330 END
```

```
LINE 1 INTERSECTS ZONE  9
LINE 1 INTERSECTS ZONE  10
LINE 1 INTERSECTS ZONE  7
LINE 1 INTERSECTS ZONE  6
LINE 1 INTERSECTS ZONE  3
LINE 3 INTERSECTS ZONE  4
LINE 3 INTERSECTS ZONE  7
LINE 3 INTERSECTS ZONE  8
LINE 3 INTERSECTS ZONE  11
LINE 5 INTERSECTS ZONE  14
LINE 7 INTERSECTS ZONE  12
LINE 7 INTERSECTS ZONE  8
LINE 9 INTERSECTS ZONE  1
LINE 9 INTERSECTS ZONE  6
LINE 9 INTERSECTS ZONE  5
02:21:54       02:21:56
```

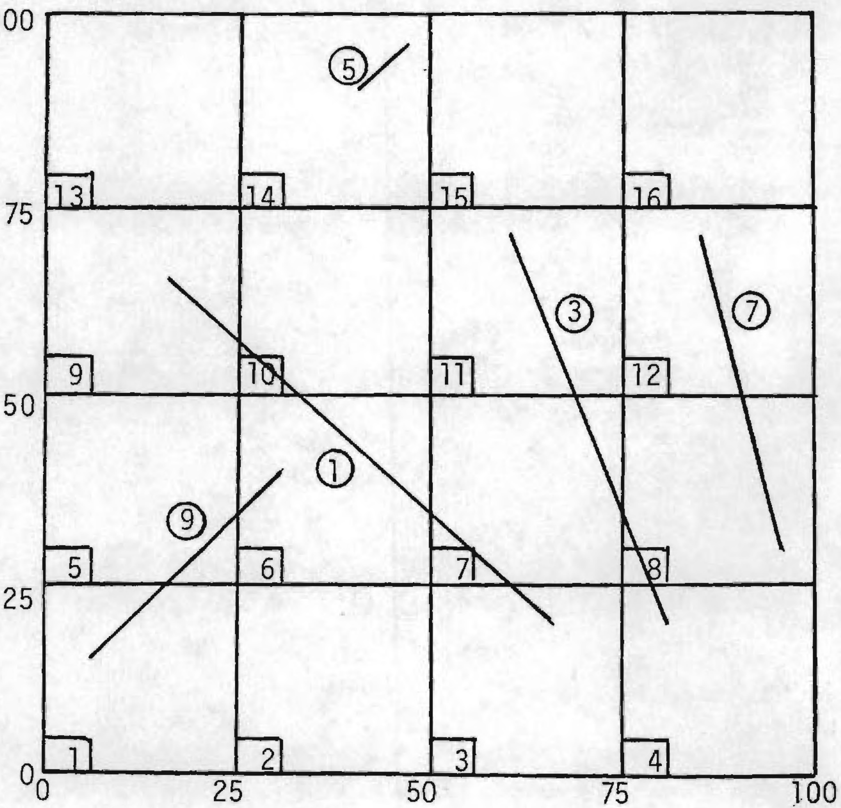


Figure 3. Sketch of BOXES Data

APPENDIX D

COMPUTER CODE FOR THE CIBLOC SUBROUTINE

Variable List

ZX(i,j); j=1,2	- (x,y) coordinates of observer
ZX(i+1,j); j=1,2	- (x,y) coordinates of target
ZZ(i,j); j=1,2	- (x,y) coordinates of building corner points
X1,Y1,Z1,X3,Y3,Z3, S1,S2,S3,S4	- see Math Model in Chapter III
ZQ(i,1)	- stores OT line of intersection
ZQ(i,2)	- stores wall of intersection
ZQ(i,j); j=3,4	- stores (x,y) coordinates of point of intersection

```
5800 '**************************************************************************
5801 '***                             CIBLOC SUBROUTINE                       ***
5802 '**************************************************************************
5810 DATA 10,7,14,11,12,13,8,9,10,7,2,5,19,15,7,18,19,6,18,9,4,3
5820 FOR I=1 TO 5
5830 FOR J=1 TO 2
5840 READ ZZ(I,J)'  READ IN BUILDING COORDINATES
5850 NEXT J
5860 NEXT I
5870 FOR I=1 TO 6
5880 FOR J=1 TO 2
5890 READ ZX(I,J)'  READ IN OBSERVER AND TARGET COORDINATES
5900 NEXT J
5910 NEXT I
5911 '**************************************************************************
5912 '***'                             MAIN PROGRAM                          ***
5913 '**************************************************************************
5920 Q=0'  SET COUNTER
5925 S$=TIME$
5930 FOR K=1 TO 6 STEP 2'  FOR EACH OBSERVER TARGET PAIR
5935 INTERS=0'  COUNTS THE NUMBER OF INTERSECTIONS FOR THIS SHOTLINE
5940 FOR L=1 TO 4'  FOR EACH SIDE OF THE BUILDING
5950 Y3=ZZ(L,2)-ZZ(L+1,2):X3=ZZ(L+1,1)-ZZ(L,1):Z3=((ZZ(L,1)*ZZ(L+1,2))-(ZZ(L+1,1
)*ZZ(L,2)))'  DEFINE CONSTANTS; SEE MATH MODEL
5960 S1=(ZX(K,1)*Y3)+(ZX(K,2)*X3)+Z3:S2=(ZX(K+1,1)*Y3)+(ZX(K+1,2)*X3)+Z3'  SEE M
ATH MODEL
5970 IF S1*S2>0 GOTO 6040' LINE SEGMENTS DO NOT INTERSECT; CHECK NEXT OT LINE
5980 Y1=ZX(K,2)-ZX(K+1,2):X1=ZX(K+1,1)-ZX(K,1):Z1=((ZX(K,1)*ZX(K+1,2))-(ZX(K+1,1
)*ZX(K,2)))'  DEFINE CONSTANTS; SEE MATH MODEL
6000 S3=(ZZ(L,1)*Y1)+(ZZ(L,2)*X1)+Z1:S4=(ZZ(L+1,1)*Y1)+(ZZ(L+1,2)*X1)+Z1'  SEE M
ATH MODEL
6010 IF S3*S4>0 GOTO 6040'  LINE SEGMENTS DO NOT INTERSECT; CHECK NEXT OT LINE
6011 IF (S1=0 AND S2=0) AND (S3=0 AND S4=0) GOTO 6040' POINTS OF INTERSECTION WI
LL BE OBTAINED FROM OTHER LINE SEGMENTS
6020 Q=Q+1' INCREMENT COUNTER
6030 INTERS=INTERS+1:ZQ(Q,1)=K:ZQ(Q,2)=L:ZQ(Q,3)=((X1*Z3)-(X3*Z1))/((Y1*X3)-(Y3*
X1)):ZQ(Q.4)=((Z1*Y3)-(Z3*Y1))/((Y1*X3)-(Y3*X1))' INCREASE THE NUMBER OF INTERSE
CTIONS:COMPUTE POINT OF INTERSECTION; STORE IN ARRAY
6035 IF INTERS>=2 GOTO 6050' AT MOST 2 POINTS OF INTERSECTION MUST BE IDENTIFIED

6040 NEXT L
6050 NEXT K
6055 F$=TIME$
6060 FOR I=1 TO Q'  PRINT RESULTS
6070 LPRINT"SHOTLINE #"ZQ(I,1)"INTERSECTS SIDE #"ZQ(I,2)"AT X="ZQ(I,3)"Y="ZQ(I,4
)
6080 NEXT I
6085 LPRINT S$,F$
6090 END
```

```
SHOTLINE # 1 INTERSECTS SIDE # 2 AT X= 13.33333 Y= 11.66667
SHOTLINE # 1 INTERSECTS SIDE # 4 AT X= 8.296296 Y= 8.703704
SHOTLINE # 3 INTERSECTS SIDE # 1 AT X= 14 Y= 11
SHOTLINE # 3 INTERSECTS SIDE # 3 AT X= 12 Y= 13
01:51:19       01:51:21
```

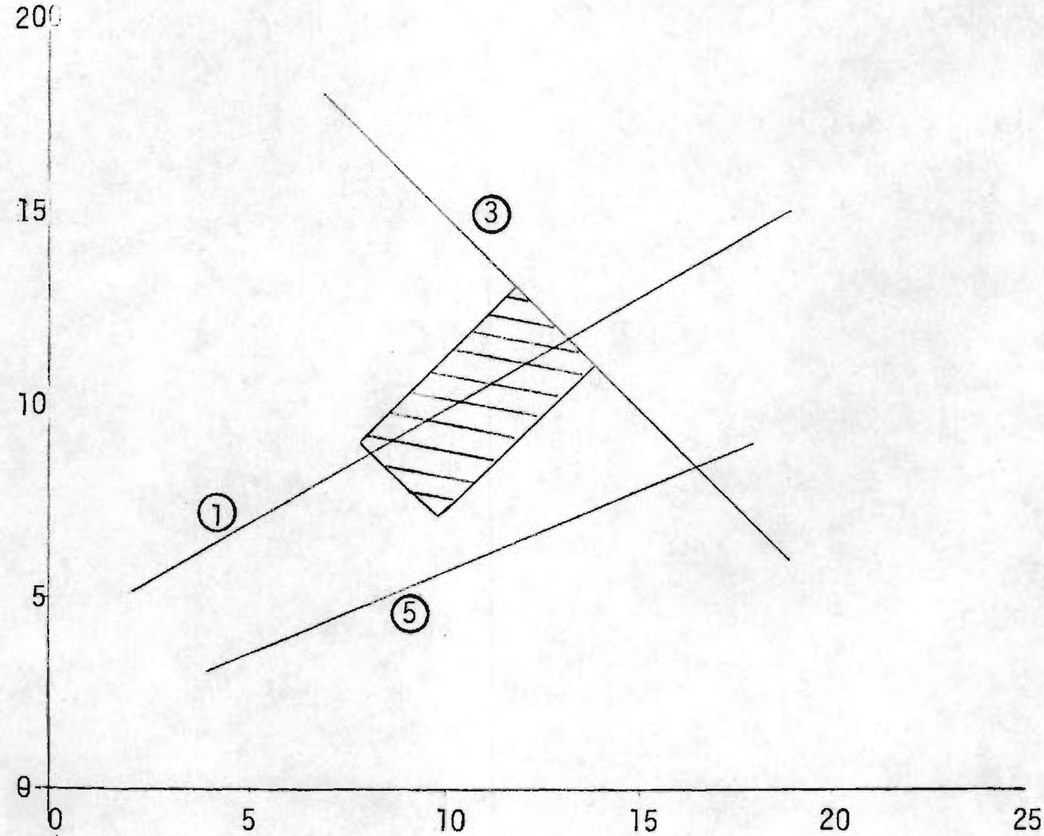


Figure 4. Sketch of CIBLOC Data

APPENDIX E

COMPUTER CODE FOR SOR SUBROUTINE

Variable List

TD(i,j); j=1,2,	- (x,y) coordinates of observer
TD(i,j); j=3,4,5,6	- (x,y) coordinates of left and right boundaries of SOR (order is irrelevant)
TT(i,j); j=1,2	- (x,y) coordinates of target
MAXD	- maximum observation range
MIND	- radius of self preservation zone
DIS	- straight line distance to target
A1,B1,C1,A2,B2, LAMDA1,LAMBDA2,X4,Y4	- see Math Model in Chapter III
TA(i,1)	- stores target number
TA(i,2)	- stores observer number

```
5400 '**************************************************************************
5401 '***                    SOR DETERMINATION SUBROUTINE                     ***
5402 '**************************************************************************
5410 DIM TA(50,2),TD(8,6),TT(6,2)
5420 DATA 5,15,1,13,3.20,8.17,1,18,7,22,13,18.12,22,6,21,14,18.12,20.17,21
5430 DATA 17,16,22,17,15,21,17,14.19,18,20,13,15,13,20,14,15,8,12,11,9,7,17,11
5440 DATA 2,19,10,30.18.25,15,18,24,13,11,3
5450 FOR I=1 TO 8
5460 FOR J=1 TO 6
5470 READ TD(I,J)
5480 NEXT J
5490 NEXT I
5492 FOR I=1 TO 6
5494 FOR J=1 TO 2
5496 READ TT(I,J)
5498 NEXT J
5499 NEXT I
5500 '*************************************************************************
5501 '***                          MAIN PROGRAM                              ***
5502 '*************************************************************************
5505 N%=1:MAXD=20:MIND=2'  SETS COUNTER,SETS MAXIMUM VISIBILITY=20,SETS SELF PRO
TECTION ZONE RADIUS = 2
5506 S$=TIME$
5510 FOR K=1 TO 6' LOOP ONCE FOR EACH OF 6 TARGETS
5520 FOR L=1 TO 8' LOOP ONCE FOR EACH OF 8 OBSERVERS
5530 DIS=SQR((TD(L,1)-TT(K,1))^2+(TD(L,2)-TT(K,2))^2)' COMPUTE DISTANCE TO TARGE
T
5540 IF DIS>=MAXD GOTO 5650' IF TARGET IS BEYOND MAXIMUM VISIBILITY CHECK NEXT T
ARGET
5550 IF DIS<=MIND GOTO 5640' IF TARGET WITHIN SELF PRESERVATION ZONE STORE IN AR
RAY
5560 A1=TD(L,3)-TD(L,1):A2=TD(L,5)-TD(L,1):B1=TD(L,4)-TD(L,2):B2=TD(L,6)-TD(L,2)
'A1 IS LEFT X COORD MINUS OBSERVER XCOORD; OTHERS ARE SIMILAR
5570 C1=(B2*A1)-(B1*A2) 'C1 IS A CONSTANT SEE MATH MODEL
5580 IF C1=0 THEN LPRINT"ERROR; CO-LONEAR BOUNNDARIES":GOTO 5650
5590 XY=TT(K,1)-TD(L,1):YY=TT(K,2)-TD(L,2) 'TRANSLATED TARGET COORDINATES
5600 LAMBDA2=((YY*A1)-(XY*B1))/C1
5610 IF LAMBDA2<0 GOTO 5650'  TARGET IS NOT IN SOR CHECK NEXT TARGET
5620 LAMBDA1=((XY*B2)-(YY*A2))/C1
5630 IF LAMBDA1<0 GOTO 5650' TARGET IS NOT IN SOR CHECK NEXT TARGET
5640 TA(N%,1)=K:TA(N%,2)=L:N%=N%+1' STORE OBSERVER AND TARGET NUMBERS IN ARRAY
5650 NEXT L
5660 NEXT K
5665 F$=TIME$
5670 FOR Q=1 TO N%-1
5680 LPRINT "TARGET #"TA(Q,1)"OBSERVED BY FIRER #"TA(Q,2)
5690 NEXT Q
5695 LPRINT S$,F$
5700 END
```

```
TARGET # 1 OBSERVED BY FIRER # 1
TARGET # 1 OBSERVED BY FIRER # 2
TARGET # 2 OBSERVED BY FIRER # 3
TARGET # 2 OBSERVED BY FIRER # 4
TARGET # 3 OBSERVED BY FIRER # 4
TARGET # 3 OBSERVED BY FIRER # 5
TARGET # 4 OBSERVED BY FIRER # 3
TARGET # 4 OBSERVED BY FIRER # 4
TARGET # 5 OBSERVED BY FIRER # 6
TARGET # 5 OBSERVED BY FIRER # 7
TARGET # 6 OBSERVED BY FIRER # 8
01:56:08        01:56:14
```

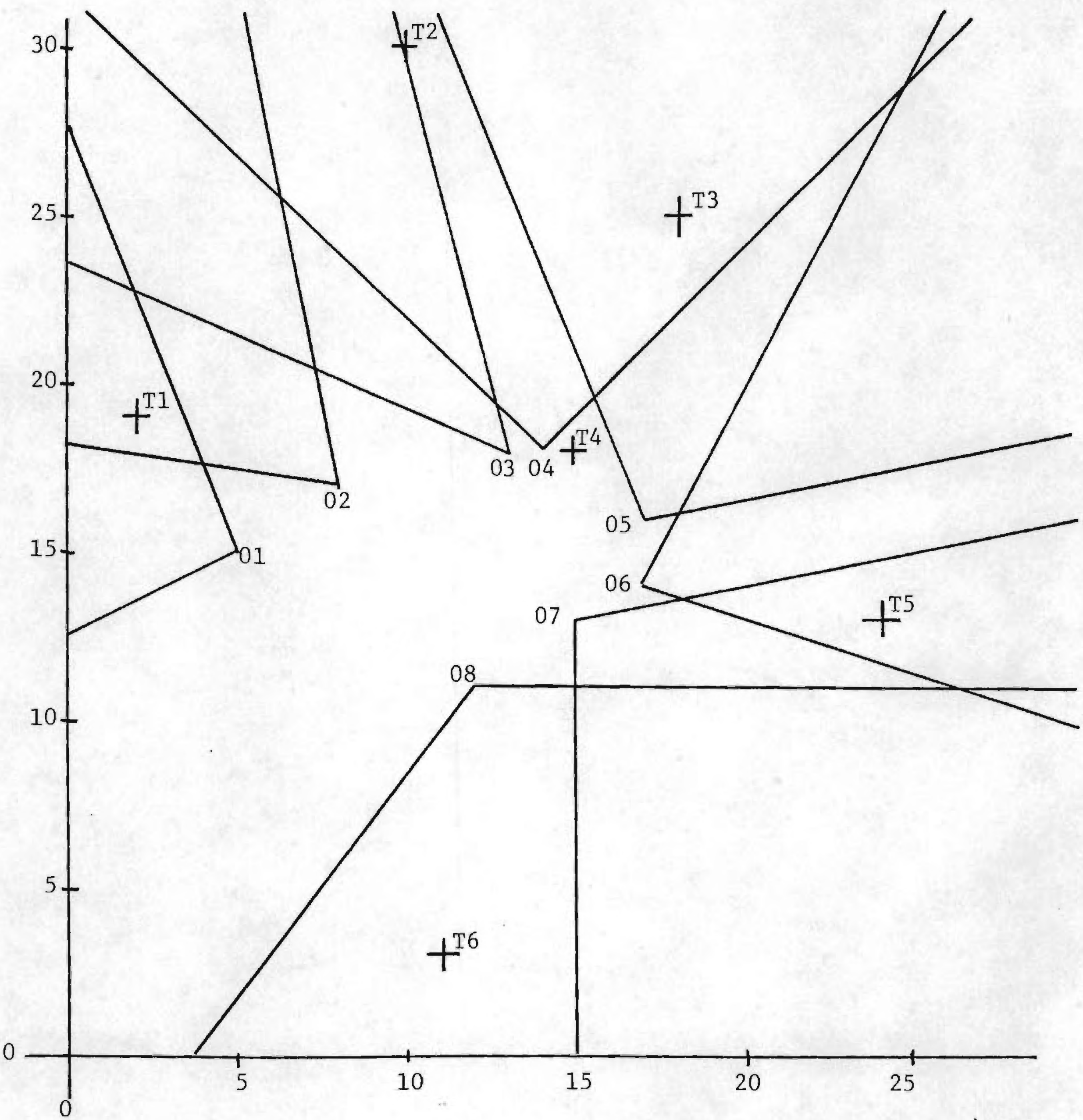


Figure 5. Sketch of SOR Data

BIBLIOGRAPHY

1. __________, "American Canadian Australian British Urban Game (ACABUG), ACABUG Concept Design, Weapons Effects, Urban Terrain and Force Organization Data Base", Technical Report 53-82, U.S. Army, TRASANA, October 1982.

2. __________, "American Canadian Australian British Urban Game (ACABUG): An Overview", Technical Report 31-83, U.S. Army TRASANA, October 1983.

3. __________,"American Canadian Australia British Urban Game (ACABUG): Systems Performance Data Development Program", Technical Report 55-82, U.S. Army TRASANA, October 1983.

4. __________, "Army Training Battle Simulation System (ARTBASS)", U.S. Army CACDA, June 1982.

5. Bazaraa, M. S. and J. J. Jarvis, Linear Programming and Network Flows, John Wiley & Sons, 1977.

6. Broome, B., "SEEFAR: An Improved Model for Producing Line-of-Sight Maps", Technical Report No. 255, U.S. Army AMSAA, September 1980.

7. __________, "BLOCKBUSTER; Basic Rules of Play; Revised", U.S. Army CACDA, 1983.

8. Callahan, L. G. and G. Crosby, "Lanchester Modeling of Small Unit Combat", Technical Report RD-CR-83-17, U.S. Army Missle Command, November 1981.

9. Callahan, L. G. and J. Loveall, "Graceful Degradation of Air Defense Capabilities", Final Summary Report, U.S. Army Missle Command, June 1982.

10. Childs, Jr., E. L. and D. D. Sipe, "MACE - A Microcomputer Based Battle Simulation", U.S. Army CAORA.

11. __________, "Combined Arms Tactical Training Simulator (CATTS): Mathematical Model User's Manual", Naval Training Equipment Center, 28 January 1977.

12. Connolly, T., Scientists, Engineers and Organizations, Brooks/Cole Engineering Division, 1983.

13. Dunnigan, J. F., The Complete Wargames Handbook, William Morrow & Company, Inc., 1980.

14. Foley, J. D. and A. Van Dam, Fundamentals of Interactive Computer Graphics, Wesley Publishing Company, 1982.

15. Hausrath, A. K., Venture Simulation in War, Business and Politics, McGraw-Hill, 1971.

16. Hoeber, F. P., Military Applications of Modeling, Gordon & Breach Science Publishers, 1981.

17. Law, A. M. and W. D. Kelton, Simulation Modeling and Analysis, McGraw-Hill Book Company, 1982.

18. Low, L. J., "Theater-Level Gaming and Analysis Workshop for Force Planning, Volumes I & II", SRI International, September 1980.

19. __________, "MICROFIX Newsletter", Volume 2, Number 7, U.S. Army FORSCOM and the Georgia Institute of Technology, December 1983.

20. __________, "Military Operations on Urbanized Terrain", FM90-10, U.S. Army, 15 August 1979.

21. Miller, R. E. and L. A. Dimillo, "Determination Whether a Point is on the Boundary of a Sector of a Circle", Technical Memorandum A-3304-214, Georgia Institute of Technology, Engineering Experiment Station, 30 June 1983.

22. __________, "Modelling, Simulation and Gaming of Warfare", Continuing Education Department, Georgia Institute of Technology, August 1983.

23. Pavlidis, T., Algorithms for Graphics and Image Processing, Computer Science Press, 1982.

24. Pritsker, A. A. B., The GASP IV Simulation Language, John Wiley & Sons, 1974.

25. Ross, S. M., Introduction to Probability Models, 2nd Edition, Academic Press, 1980.

26. Sharp, D. A., "A New Movement and Terrain Representation System", Proceedings of the Twenty Second Annual U.S. Army Operations Research Symposium, October 1983.